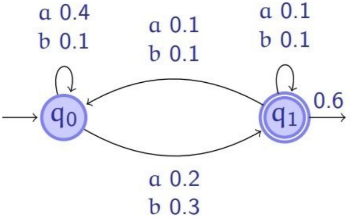
Tensor Product Representations of Subregular Formal Languages

Jon Rawski
Dept. of Linguistics
Institute for Advanced Computational Science
Stony Brook University

Regular Languages & Finite-State Automata

Regular Expressions, weighted FSA, finite monoid, etc.



Operator Representation

$$\boldsymbol{\alpha} = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix} \quad \mathbf{A}^{a} = \begin{bmatrix} 0.4 & 0.2 \\ 0.1 & 0.1 \end{bmatrix}$$

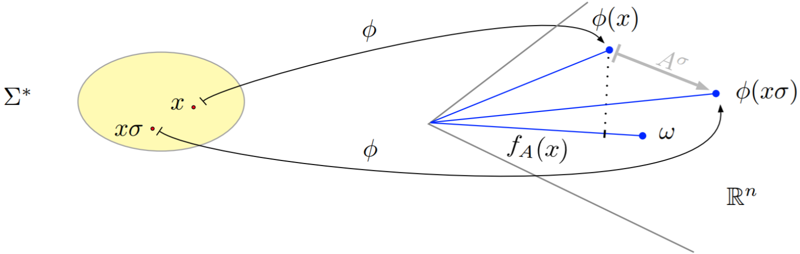
$$\boldsymbol{\omega} = \begin{bmatrix} 0.0 \\ 0.6 \end{bmatrix} \quad \mathbf{A}^{b} = \begin{bmatrix} 0.1 & 0.3 \\ 0.1 & 0.1 \end{bmatrix}$$

$$f(ab) = 0.4 \times 0.3 \times 0.6 + 0.2 \times 0.1 \times 0.6 = 0.084$$
$$= \boldsymbol{\alpha}^{\top} \mathbf{A}^{a} \mathbf{A}^{b} \boldsymbol{\omega}$$

p.c.

Guillaume Rabusseau

Finite-State Automata & Representation Learning



An FSA induces a mapping $\phi : \Sigma^* \to \mathbb{R}$

The mapping ϕ is compositional

The output $f_A(x) = \langle \phi(x), \omega \rangle$ is linear in $\phi(x)$

p.c. Guillaume Rabusseau

Finite-State Automata Are Everywhere

image processing (Kari, 1993).

automatic speech recognition (MM, Pereira, Riley, 1996, 2008).

speech synthesis (Sproat, 1995; Allauzen, MM, Riley 2004).

machine translation (e.g., Iglesias et al., 2011).

many other NLP tasks (very long list of refs).

bioinformatics (Durbin et al., 1998).

optical character recognition (Bruel, 2008).

model checking (Baier et al., 2009; Aminof et al., 2011).

machine learning (Cortes, Kuznetsov, MM, Warmuth, 2015).

Neural Nets & Regular Languages

Kleene 1956: Regular Expressions = generalization of NN behavior
Giles et al 1992, Avcu et al 2018: RNNs learning regular languages
Weiss et al., 2018, Ayache et al., 2018: extracting FSA from RNNs
Rabusseau et al 2019: linear-2 RNNs = weighted DFA
Merrill 2019: Sequential NN are subregular automata
McCoy et al 2019: RNNs Implicitly Implement Tensor Product Representations

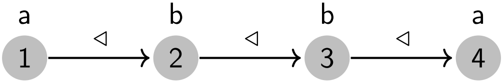
Finite Model Theory

'word' is synonymous with 'structure.'

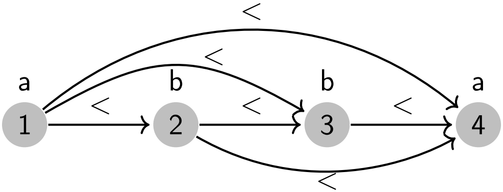
- A model of a word is a representation of it.
- A (Relational) Model contains two kinds of elements.
  - A domain: a finite set of elements.
  - Relations over domain elements.
- Every word has a model.
- Different words have different models.
- general: strings, infinite strings, trees, texts, graphs, hypergraphs, etc

Finite Word Models

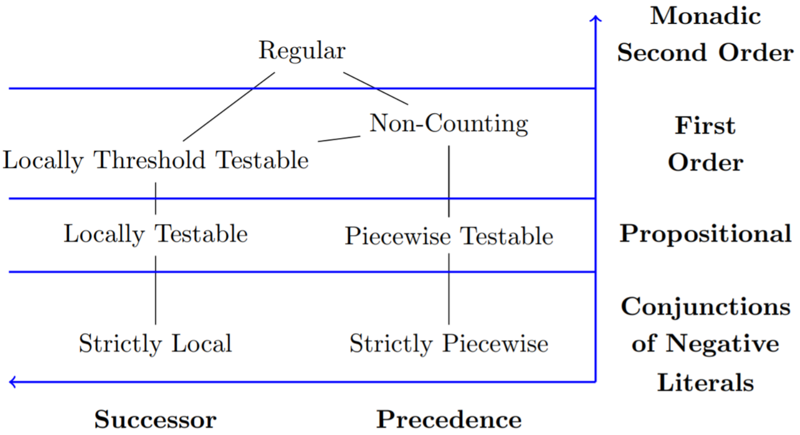
1. Successor (Immediate Precedence)



2. General precedence



Subregular Hierarchy (Rogers et al 2013)



Tensors: Quick and Dirty Overview

Order 1 — vector:

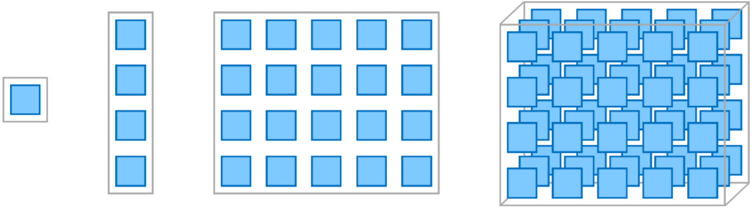
$$\vec{v} \in A = \sum_i C_i^v \vec{a_i}$$

Order 2 — matrix:

$$M \in A \otimes B = \sum_{ij} C_{ij}^M \vec{a_i} \otimes \vec{b_j}$$

Order 3 — Cuboid:

$$R \in A \otimes B \otimes C = \sum_{ijk} C_{ijk}^R \vec{a_i} \otimes \vec{b_j} \otimes \vec{c_k}$$



Tensors: Quick and Dirty Overview

Tensor contractions:

Order 1 $\times$ order 1: inner product (dot product)

Order 2 $\times$ order 1: matrix-vector multiplication

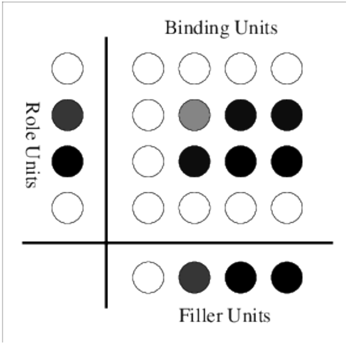
Order 2 $\times$ order 2: matrix multiplication

Tensor contraction is nothing fancier than a generalization of these operations to any order.

Order n $\times$ order m: sum through shared indices.

Order n $\times$ order m contraction yields tensor of order $n+m-2$.

Tensor Product Representations (Smolensky 1990)

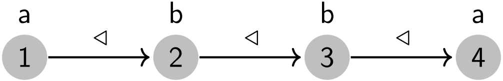


Crucial to dynamical systems models of linguistic cognition

- beim Graben and Gerth: parsing with tensor products of tree languages
- beim Graben et al: EEG dynamics via tensor product parsing
- Hale and Smolensky: CFGs over recursive tree tensors
- Smolensky: Language as optimization over string tensors

Embedding the Model: Domain

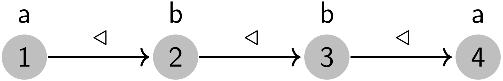
The set of one-hot vectors in $D \cong \mathbb{R}^{|D|}$ models the logical atoms of D. For Example:



$$D = \{1,2,3,4\} \Rightarrow \mathbf{1} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \mathbf{2} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{3} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \mathbf{4} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Embedding the Model: Relations

A k-ary relation r in M is computed by an order-k tensor $\mathcal{R} = \{r_{i_1,\ldots,i_k}\}$, whose truth value $[[r(e_{i_1},\ldots,e_{i_k})]] = \mathcal{R}(\mathbf{e}_{i_1},\ldots,\mathbf{e}_{i_k})$



$$\mathcal{R}_a = \begin{bmatrix}1\\0\\0\\1\end{bmatrix} \mathcal{R}_b = \begin{bmatrix}0\\1\\1\\0\end{bmatrix} \mathcal{R}_{\triangleleft} = \begin{bmatrix}0 & 1 & 0 & 0\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\\0 & 0 & 0 & 0\end{bmatrix}$$

Tensors as Functions

Tensor-linear map isomorphism (Bourbaki, 1985; Lee, 1997)

For any multilinear map $f : V_1 \to \ldots \to V_n$ there is a tensor $T^f \in V_n \otimes \ldots \otimes V_1$ such that for any $\overrightarrow{v_1} \in V_1, \ldots, \vec{v}_{n-1} \in V_{n-1}$, the following equality holds

$$f(\overrightarrow{v_1}, \ldots, \overrightarrow{v_{n-1}}) = T^f \times \overrightarrow{v_1} \times \ldots \times \overrightarrow{v_{n-1}}$$

Tensors therefore act as functions, with tensor contraction as function application.

Properties of Linear Maps propagate to tensors

Logical Formulas (Sato 2017)

$$[[\neg F]]' = 1 - [[F]]'$$

$$[[F_1 \wedge \cdots \wedge F_h]]' = [[F_1]]' \cdots [[F_h]]'$$

$$[[F_1 \vee \cdots \vee F_h]]' = \min_1([[F_1]]' + \ldots + [[F_h]]')$$

$$[[\exists y F]]' = \min_1(\sum_{i=1}^{N} [[F_{y \leftarrow e_i}]]')$$

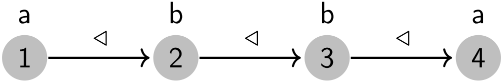
$$[[\forall y F]]' = 1 - \min_1(\sum_{i=1}^{N} 1 - [[F_{y \leftarrow e_i}]]')$$

Here the operation $\min_1(x) = \min(x, 1) = x$ if $x < 1$, otherwise 1, as componentwise application.

FO(◁) = LTT Example: Exactly 1 b

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

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



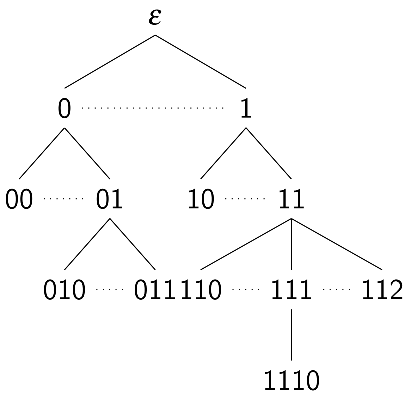
FO(<) = Star-Free Example: No 2 l's but allow lrl

a.	navalis	'naval'
b.	solaris	'solar' (*solalis)
c.	lunaris	'lunar' (*lunalis)
d.	litoralis	'of the shore'

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

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

Extension 1: Tree Models (Rogers 2003)



Extension 2: First-Order Transductions (Courcelle 2001)

$$\begin{aligned}
\mathtt{a}^{O}(x) &\stackrel{\text{def}}{=} \mathtt{a}(x) \vee \mathtt{b}(x)\\
\mathtt{b}^{O}(x) &\stackrel{\text{def}}{=} \mathtt{FALSE}\\
\mathtt{p}^{O}(x) &\stackrel{\text{def}}{=} \mathtt{p}(x)\\
\mathtt{s}^{O}(x) &\stackrel{\text{def}}{=} \mathtt{s}(x)\\
lic^{O}(x) &\stackrel{\text{def}}{=} \mathtt{TRUE}
\end{aligned}$$