Towards a Theory of Information Flow in the Finitary Process Soup

Spencer Mathews
smathews@cse.ucdavis.edu
Department of Computer Science
Complexity Sciences Center
The University of California at Davis

June 9, 2010

Abstract

The Finitary Process Soup is a model of evolutionary self-organization. The elementary particles of this system, ϵ-transducers, possess well defined mathematical properties, thus permitting structural analysis on multiple scales. ϵ-Transducers are embodiments of communication channels, and their interactions occur via functional composition. We thus explore the notion of information as currency in this system. We characterize ϵ-transducers through their roles as channels, measuring their channel capacity. We conclude that transducer composition does can not increase channel capacity. We ask if, and how, channel capacity may be used to understand the roles of transducers in emergent autocatalytic networks, and arrive at the surprising result that population dynamics favors transducers with channel capacity of zero.

Contents

1 Introduction

The broader context for this work is within the population dynamics of the Finitary Process Soup [2]. This evolutionary model, introduced below, enables a mathematically tractable analysis of hierarchical self-organization. In order to better understand the forces which drive self-organization, we seek to devise a categorization of the first order elements within this model. The hope is that we may begin to speak more broadly about classes of objects, and interactions among these classes, instead of among individual objects themselves — a dimensionality reduction, of sorts. This categorization will help us to better understand the roles of individuals within the Finitary Process Soup.

This project is but a step toward the larger goal of extending Computational Mechanics to ϵ-transducers and to explore the notion of information flow in evolving networks of these machines. Prior to addressing flow in networks, we must first quantify information flow in individual transducers.

ϵ-Transducers may be viewed through their dual roles as both communication channels and functional mappings capable of transforming the structure of their input. We will focus primarily on the former, and characterize ϵ-transducers in the one state library in terms of their *channel capacities*, as well as on their input and output languages. The impact of transducer composition on channel capacity will be also examined.

We find that single state ϵ-transducers embody a diverse set of communication channels. The measure of channel capacity follows a certain algebra over the composition operation. Surprisingly, however, a machine's persistence in an evolving population varies inversely with channel capacity.

2 Background

Before we report our results, some preliminary material will be reviewed. We will first describe what we mean by a communication channel, and how we can measure its capacity. Then, we will introduce ϵ-transducers as our primary objets. We will go on to show how ϵ-transducers are naturally viewed as communication channels. In the next section we will introduce dynamics over transducers, and introduce the Finitary Process Soup more completely.

2.1 Discrete Channels

We define a *discrete channel* [1] to be a system consisting of an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$, and a probability transition matrix $p(Y|X)$ that expresses the probability of observing the output symbol y given that we send the symbol x. $\mathcal{X}$ and $\mathcal{Y}$ are finite sets, and entries in the matrix are conditional probabilities. For every x and y, $p(y|x) \geq 0$ and for every x $\sum_y p(y|x) = 1$.

We commonly deal with *memoryless* channels, in which the probability distribution of the output depends only on the input at that time and is conditionally independent of previous channel inputs or outputs.

2.1.1 Channel Capacity

The *channel capacity* [1] of a discrete memoryless channel is:

$$C = \max_{p(x)} I(X;Y) \tag{1}$$

This capacity specifies the highest rate, in bits, at which information may be reliably transmitted through the channel, and has the following properties:

$C \geq 0$
$C \leq \log |\mathcal{X}|$ since $C = \max\ I(X;Y) \leq \max\ H(X) = \log |\mathcal{X}|$
$C \leq \log |\mathcal{Y}|$

From its definition we can see that channel capacity is an optimization problem over *mutual information*. The most natural formulation for our purposes is:

$$I(X;Y) = H[Y] - H[Y|X] \tag{2}$$

$I(X;Y)$ is a symmetric quantity, and a continuous function of $p(x)$, although in general there is no closed form solution for channel capacity.

2.2 ϵ-Transducers

We now move on to define the objects of our analysis, ϵ-transducers, and we do so in terms of ϵ-machines. ϵ-Machines are minimal and maximally predictive models of causal processes [3]. An ϵ-machine is defined as $M = \{\boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{T}}\}$,

where $\boldsymbol{\mathcal{S}}$ is a set of *causal states* and $\boldsymbol{\mathcal{T}}$ is the set of *transitions* between them, with $M_{ij}^{(s)},\ s \in \mathcal{A}$. An ϵ-machines recurrent states form a single *strongly connected* component, and transitions are *unifilar* (i.e. deterministic).

We formalize an ϵ-transducer as a reinterpretation of an ϵ-machine. It is common to consider ϵ-machines over a binary alphabet. We consider an ϵ-transducer to be an ϵ-machine over an extended alphabet of size $|\mathcal{A}|^2$. We interpret the symbols labeling the transitions in the alphabet $\mathcal{A}$ as consisting of two parts: an *input symbol* that determines which transition to take from a state and an *output symbol* which is emitted on taking that transition.

We can think of transducers as implementing functional mappings on several levels of interpretation. They map an input character to an output character, as well as input strings to output strings. Associated with each ϵ-transducer is a set of all possible input strings, its *input language* $\mathcal{L}_{in}$, and a set of all possible output strings, its *output language* $\mathcal{L}_{out}$. The transducer specifies a mapping between these sets, and is thus a compact representation

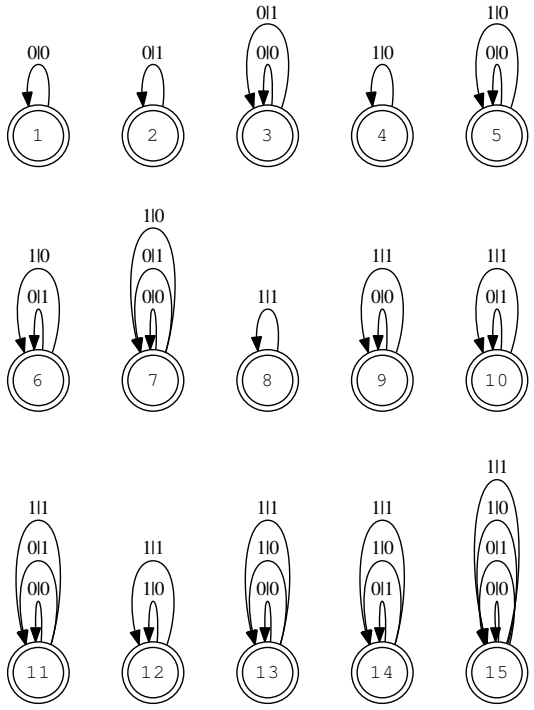

Figure 1: The library of single state ϵ-machines

of a mapping from one formal language to another.

As we continue in our discussion we will use the terms ϵ-transducer, ϵ-machine, transducer, and machine interchangeably, in all cases referring to ϵ-transducers.

The set of single state ϵ-transducers is illustrated in figure 1 .

2.2.1 Composition

Transducer composition provides a way of functionally coupling multiple machines, and is essential to the population dynamic described in the next section. Composition creates a new mapping wherein the output string from one machine T_i is taken as the input language of another T_j. The resulting string reflects the combined processing of both transducers. If the machine which implements this mapping is T_k, we write $T_k = T_j \circ T_i$. It should be noted that composition is not a symmetric operation..

The system of compositions between machines may be described with a set of *interaction matrices*, $\mathcal{G}^{(k)}$.

$$\mathcal{G}_{ij}^{(k)} = \begin{cases} 1 & \text{if } T_k = T_j \circ T_i \\ 0 & \text{otherwise} \end{cases}$$

A more compact description, requiring only a single matrix $\mathcal{G}$, is given as:

$$\mathcal{G}_{ij} = T_k \text{ where } T_k = T_j \circ T_i$$

$$\begin{pmatrix}
T_1 & T_2 & T_3 & T_0 & T_1 & T_2 & T_3 & T_0 & T_1 & T_2 & T_3 & T_0 & T_1 & T_2 & T_3 \\
T_0 & T_0 & T_0 & T_1 & T_1 & T_1 & T_1 & T_2 & T_2 & T_2 & T_2 & T_3 & T_3 & T_3 & T_3 \\
T_1 & T_2 & T_3 & T_1 & T_1 & T_3 & T_3 & T_2 & T_3 & T_2 & T_3 & T_3 & T_3 & T_3 & T_3 \\
T_4 & T_8 & T_{12} & T_0 & T_4 & T_8 & T_{12} & T_0 & T_4 & T_8 & T_{12} & T_0 & T_4 & T_8 & T_{12} \\
T_5 & T_{10} & T_{15} & T_0 & T_5 & T_{10} & T_{15} & T_0 & T_5 & T_{10} & T_{15} & T_0 & T_5 & T_{10} & T_{15} \\
T_4 & T_8 & T_{12} & T_1 & T_5 & T_9 & T_{13} & T_2 & T_6 & T_{10} & T_{14} & T_3 & T_7 & T_{11} & T_{15} \\
T_5 & T_{10} & T_{15} & T_1 & T_5 & T_{11} & T_{15} & T_2 & T_7 & T_{10} & T_{15} & T_3 & T_7 & T_{11} & T_{15} \\
T_0 & T_0 & T_0 & T_4 & T_4 & T_4 & T_4 & T_8 & T_8 & T_8 & T_8 & T_{12} & T_{12} & T_{12} & T_{12} \\
T_1 & T_2 & T_3 & T_4 & T_5 & T_6 & T_7 & T_8 & T_9 & T_{10} & T_{11} & T_{12} & T_{13} & T_{14} & T_{15} \\
T_0 & T_0 & T_0 & T_5 & T_5 & T_5 & T_5 & T_{10} & T_{10} & T_{10} & T_{10} & T_{15} & T_{15} & T_{15} & T_{15} \\
T_1 & T_2 & T_3 & T_5 & T_5 & T_7 & T_7 & T_{10} & T_{11} & T_{10} & T_{11} & T_{15} & T_{15} & T_{15} & T_{15} \\
T_4 & T_8 & T_{12} & T_4 & T_4 & T_{12} & T_{12} & T_8 & T_{12} & T_8 & T_{12} & T_{12} & T_{12} & T_{12} & T_{12} \\
T_5 & T_{10} & T_{15} & T_4 & T_5 & T_{14} & T_{15} & T_8 & T_{13} & T_{10} & T_{15} & T_{12} & T_{13} & T_{14} & T_{15} \\
T_4 & T_8 & T_{12} & T_5 & T_5 & T_{13} & T_{13} & T_{10} & T_{14} & T_{10} & T_{14} & T_{15} & T_{15} & T_{15} & T_{15} \\
T_5 & T_{10} & T_{15} & T_5 & T_5 & T_{15} & T_{15} & T_{10} & T_{15} & T_{10} & T_{15} & T_{15} & T_{15} & T_{15} & T_{15}
\end{pmatrix}$$

Figure 2: Interaction matrix for the single state transducers.

The transition matrix which describes compositions among single state ϵ-transducers is shown in figure 2. A property unique to the set of single state ϵ-transducers is that it is closed under composition.

2.3 Transducers as Channels

The set of single state ϵ-transducers are realizations of discrete, memoryless, binary channels. Recall that, by definition, ϵ-transducers are unifilar over their $input|output$ pairs. That is, they need not be unifilar with respect to their input or output alphabets alone, only over the Cartesian product of these alphabets. This potential nondeterminism leads some ϵ-transducers to behave as noisy channels.

Transition probabilities play an essential role in the functioning of ϵ-machines proper. In our formulation of ϵ-transducers we neglected to clarify the role of probability. In transducers which happen to be unifilar over their input alphabet, probability is largely irrelevant since the input symbol plays an equivalent role in specifying a transition. Probabilities only really assume meaning when multiple outputs are possible given a single input. In this case output symbols become a stochastic function of input. For simplicity, we assume that the outputs are are all equally likely.

3 Dynamical System

Our system of focus is the Finitary Process Soup (FPS). The soup consists of a population P of N ϵ-machines, and evolves through discrete replication steps. The system's large state space is described by N, the population size, and a vector $\mathbf{p}$ of length N, where p_i is the fraction of machine type i in the population at a given time.

3.1 Soup Dynamics

The population dynamics of the Finitary Process Soup unfold in discrete time steps. Interactions occur through composition, and in this way we conceive of transducers transforming one another in the soup. A single replication is determined through composition and replacement in a two-step sequence:

1. Construct ϵ-machine T_C by forming the composition $T_C = T_B \circ T_A$ from T_A and T_B randomly selected from the population and minimizing.

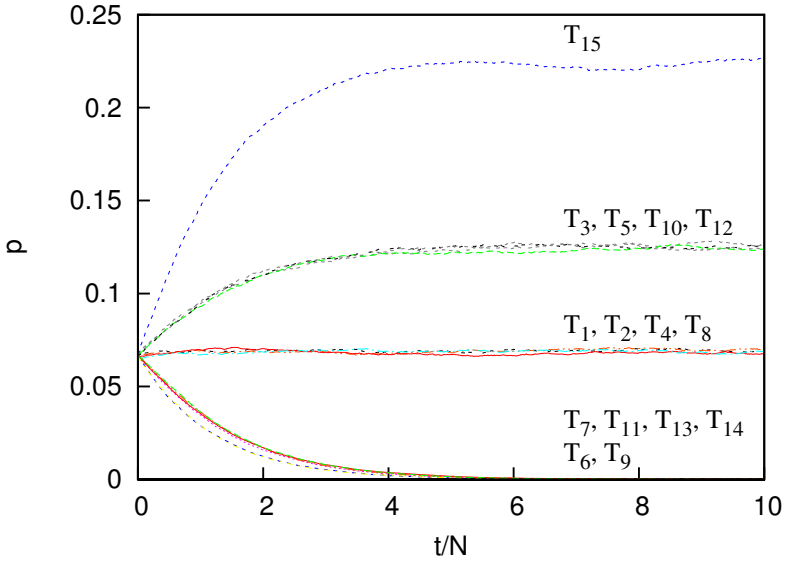

Figure 3: Population dynamics of single state ϵ-machines starting from a uniform distribution and population size 100,000.

2. Replace a randomly selected ϵ-machine, T_D, with T_C.

The population dynamics of a typical run of the single state is shown in in figure 3. The relative abundance of each transducer type changes over time until a steady state is reached.

3.2 Meta-Machines

While the FPS state space is large, the trajectory of a given population through this state space tends to be quite specific. Population dynamics lead it to spontaneously self-organize into stabilized auto-catalytic networks. These networks reflect sets of ϵ-machines whose members recursively participate in their own production.

A *meta-machine* captures the notion of invariant set, $\Omega = \mathcal{G} \circ \Omega$, and is a set of machines that is closed and self-maintained under composition. $\Omega \subseteq P$ is a meta-machine if and only if

(i) $T_i \circ T_j \in \Omega$, for all $T_i, T_j \in \Omega$ and

(ii) For all $T_k \in \Omega$, there exists $T_i, T_j \in \Omega$, such that $T_k = T_i \circ T_j$.

The 9 transducers which persist in figure 3 form such a set in the single state soup. Meta-machines are an emergent second order structure in the Finitary Process Soup, and a significant goal of this project to understand and characterize the forces which drive this spontaneous ordering.

4 Methods

The tasks of partitioning the set of ϵ-transducers and computing channel capacity were both accomplished through the development of Python code. Each ϵ-transducer was read in from a file and converted to a set of four transition matrices — one for each *input*|*output* pair. Matrix M^0 corresponds to 0|0, M^1 to 0|1, M^2 to 1|0, and M^3 to 1|1. Transition "probabilities" were assigned by assuming a uniform probability over outgoing transitions.

A transducer's input and output languages may be analyzed separately by creating a new set of matrices concerned only with the respective alphabet. We do this by summing the above mentioned matrices with shared input or output symbols. For example, $M^0_{in} = M^0 + M^1$ and $M^1_{out} = M^1 + M^3$.

These matrices may be used to determine if a transducer accepts both 0 and 1 as input, and whether both 0 and 1 may be observed as output for some input. Since channel capacity reflects a maximum over all possible input distributions, if only a single symbols is acceptable as input there is only one such distribution, no uncertainty, and thus no way for the channel capacity to be anything but zero. When both 0 and 1 occur as possible input symbols to a transducer, we say that $\mathcal{L}_{in} = \Sigma^*$. We define the class Σ^*_{in} as the set containing all such transducers.

In order for a channel to have a non-zero channel capacity there must also be some potential uncertainty regarding its outputs. Practically speaking, this means that 0 and 1 must both be present as the output symbol on some transition. When both 0 and 1 occur as output symbols, we say that $\mathcal{L}_{out} = \Sigma^*$. We define the class Σ^*_{out} as the set containing all such transducers.

We define the class Σ^* to be the set containing all transducers for which all symbols are possible as both input and output. Thus, $\Sigma^* = \Sigma^*_{in} \cap \Sigma^*_{out}$.

The task of computing the channel capacity involves finding the maximum mutual information over all input distribution. Each distribution requires a new conditional probability matrix to be computed for each transducer. For a binary channel, the input distribution varies with a single parameter p,

where $P(X = 1) = p$ and $P(X = 0) = 1 - p$.

The values of $H[X]$ and $H[Y|X]$ required to compute $I(X;Y)$ both vary with p, and are calculated from this conditional probability matrix.

Given the input distribution and the conditional matrix of the channel, we may compute the marginal distribution of outputs through matrix multiplication as

$$P(Y) = \hat{P}(X)[P(Y|X)]$$

where $\hat{P}(X)$ is the probability distribution over inputs written as a row vector, and $[P(Y|X)]$ is the conditional matrix described above.

The conditional entropy $H[Y|X]$ is computed using the formula

$$H[Y|X] = \sum_{x \in \mathcal{X}} p(x) H[Y|X = x]$$

Python code as written to compute the mutual information for values of p in the unit interval, in increments of 0.01.

5 Results

We present the results of our partitioning effort, first done in terms of languages, and then refined by channel capacity. The combination of the two yield insight into the dynamics which drive the single state soup to its steady state meta-machine.

5.1 Partitioning

As a first order of business before computing channel capacity, we partition the set of single state ϵ-transducers, based on their input and output languages languages, into Σ^*_{in},Σ^*_{out}, Σ^*. Transducers which only have single arcs, and consequently fall into none of these categories, form an additional class. Each machine type is associated with one of these sets:

$$\begin{aligned} 5, 6, 7, 9, 10, 11, 13, 14, 15 &\in \Sigma^*_{in} \\ 3, 6, 7, 9, 11, 12, 13, 14, 15 &\in \Sigma^*_{out} \\ 6, 7, 9, 11, 13, 14, 15 &\in \Sigma^* \\ 1, 2, 4, 8 &\in \text{`single arcs'} \end{aligned}$$

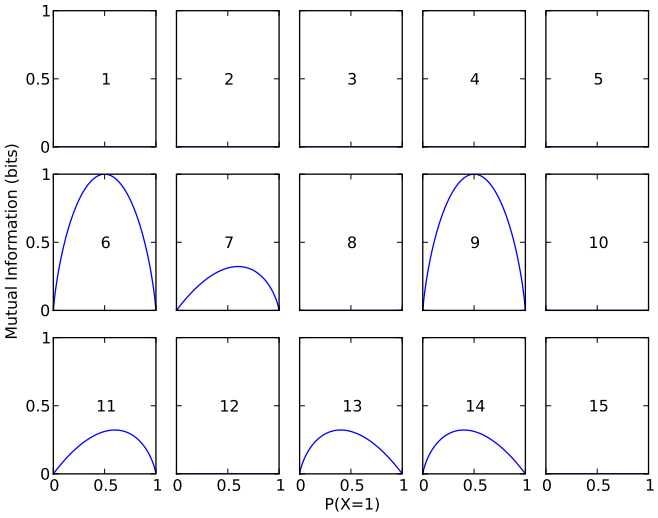

Figure 4: Mutual information as a function of p(X) of single state transducers.

There are several things to notice. As illustrated in figure 1, there is a large degree of symmetry among machines. For each machine, with the exception of machine 15, there is a topologically equivalent transducer differing only in that 0s and 1s are reversed. As we continue on to measure channel capacity, we note that Σ^* is the only class where channel capacity may be nonzero.

5.2 Channel Capacities

We summarize our channel capacity results in figure 4. The plots illustrate how the mutual information between input and output varies with input distribution, with channel capacity being found at the maxima.

Quantitative results are summarized in table 1. Channel capacity values, and the probability distributions at which they occur, induce a refinement to our partitioning based on language.

All transducers which are not in the class Σ^* have a channel capacity of zero. All transducers in Σ^*, with the exception of machine 15, have a positive channel capacity. We define this class as $\Sigma^{*'}$, where $\Sigma^{*'} = \Sigma^* \backslash T_{15}$. As a channel with too much noise to transmit any useful information, machine 15

Machine Number	Channel Capacity	P(X=1)
6	1.0	0.5
7	0.3219	0.6
9	1.0	0.5
11	0.3219	0.6
13	0.3219	0.4
14	0.3219	0.4

Table 1: Single state transducers with non-zero channel capacities.

defines its own class.

Within $\Sigma^{*'}$ we may specify a further refinement. Machine 9 is the prototypical discrete noiseless channel. Machine 6 flips input bits with perfect fidelity, and may be interpreted as either a noiseless channel, or as binary symmetric channel with a 100% error rate. Both of these transducers share a mutual information curve (figure 5), and maximize channel capacity. They are bitwise symmetric.

Transducers 7, 11, 13 and 14 share the common feature that one of their input symbols transmits noiselessly, while the other is noisy in the sense that it can result in multiple output symbols. We refer to these machines as having a *reduced capacity*, since the channel capacities of these transducers is only 0.3219. Machines 7 and 11 (figure 6) all share their peak when $P(X = 1) = 0.6$, whereas machines 13 and 14 (figure 7) both peak at 0.4. The transducers in each pair have identical inputs, but their output bits are flipped (transducers 7 and 14, 11 and 13 are topologically equivalent with respect to their transitions, with the exception that all of their bits are flipped).

In the case of these reduced capacity transducers, the maximal mutual information of 0.3219 occurs at a distribution biased toward the noiseless input. This leads to an output distribution of $(0.8, 0.2)$ in favor of the output symbol yielded by noiseless transmission. Thus $H[X] = H(0.8) = 0.7219$ and $H[Y|X] = \frac{2}{5}$.

It should be noted that Shannon's Second Theorem states that it is possible for a channel to achieve its channel capacity in practice. How to do this is not always clear. The problem of constructing optimal codes for these noisy transducers is an interesting one, but beyond the scope of this paper.

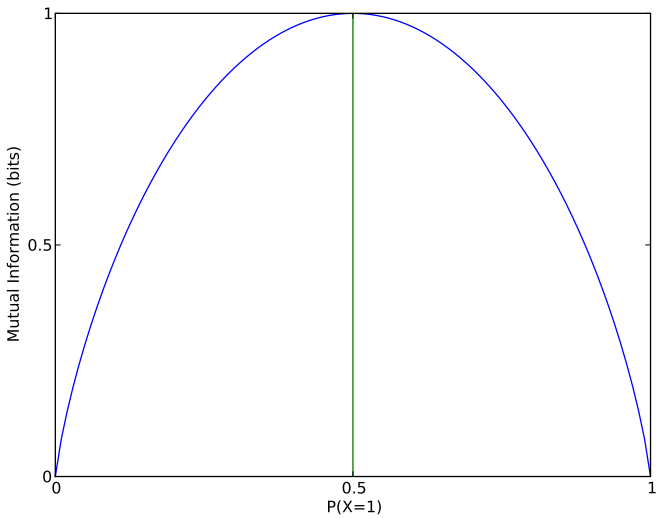

Figure 5: Mutual information as a function of p(X) for transducers 6 and 9.

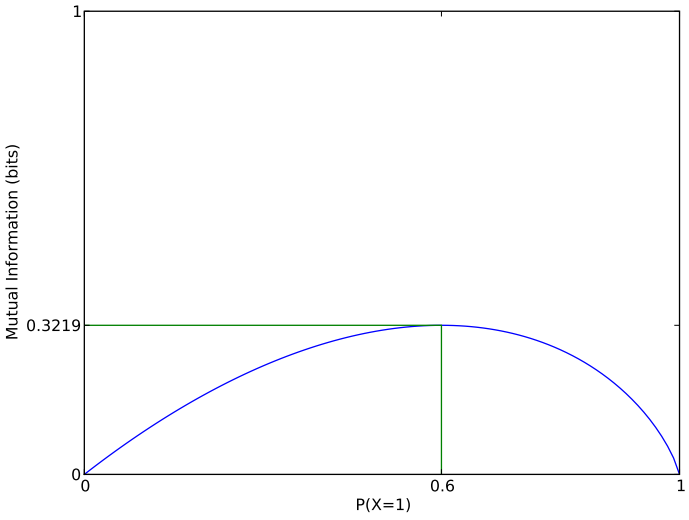

Figure 6: Mutual information as a function of p(X) for transducers 7 and 11.

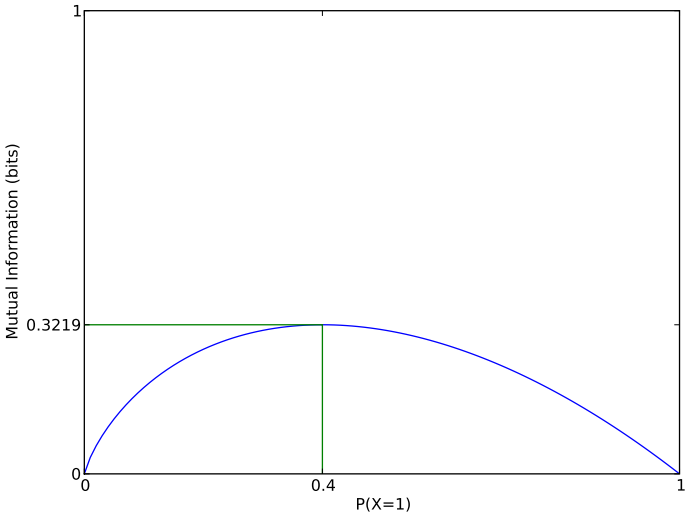

Figure 7: Mutual information as a function of p(X) for transducers 13 and 15.

5.2.1 Composition

We investigate the interaction of channel capacity and composition. Specifically, given two transducers of known channel capacity, we would like to predict the channel capacity of their composition product. We find a regularity in these interactions.

When transducers 6 and 9, those with channel capacity of 1, are self composed or composed with each other, the resulting machine also has channel capacity 1. Composition within this set of machine causes it to map back on itself. When one of these two machines is composed with one of the reduced capacity machines, in either direction, the result is another reduced capacity machine. When any of the reduced capacity machines is composed to another from this set, the product is another reduced capacity machine or machine 15, which has a capacity of zero.

Thus, the composition operation over the single state machines never increases channel capacity. When two ϵ-transducers are composed to produce a new transducer, the composition product will have a channel capacity either the same or less than the parent with the least channel capacity.

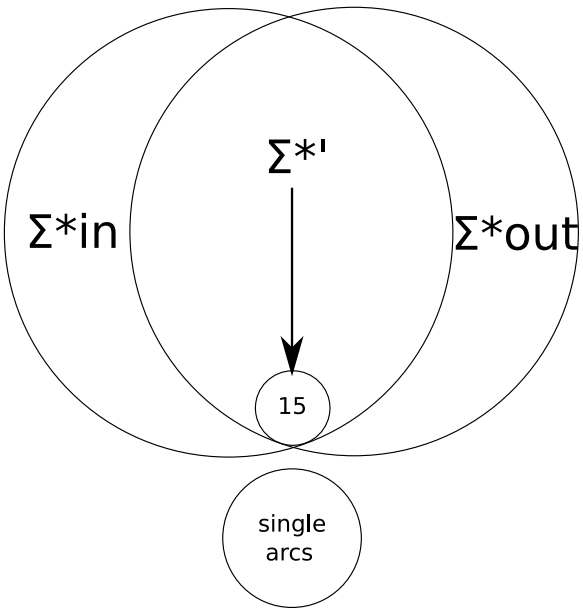

Figure 8: Partitioning based on input/output language and channel capacity.

5.3 Insights into Dynamics

The categories induced by differentiating among input and output language, further refined by channel capacity, provide a coherent framework by which to view the population dynamics of the single state soup. The categories which are most useful for analysis are: (i) those with single arcs (ii) Σ^*_{in} (iii) Σ^*_{out} (iv) T_{15} and (v) $\Sigma^{*'}$. For convenience we redefine our notation, and consider Σ^*_{in} and Σ^*_{out} to only contain transducers not in Σ^*. The purpose is to create disjoint sets.

The first thing to notice about this categorization is that all of these classes, with the exception of $\Sigma^{*'}$, map completely back onto themselves. Furthermore, they technically fulfill the definition of a meta-machine. Certain groupings of these sets are also closed and self maintained. Further experimentation is required to determine if a population containing just these subsets will persist, or whether the dynamics will lead to the elimination of some machines. The result is that it may be necessary to develop our notion of meta-machine further by integrating a notion of stability.

The set of transducer in $\Sigma^{*'}$, those with positive channel capacity, form a unique set for the purposes of analysis. The set is self reproducing and, addi-

tionally, they produce machine 15, making them what we might term a *leaky* meta-machine. Unlike any of the other partitions, they are not generated through the interaction of any transducers not in that set, so it is perhaps not surprising that they are eventually eliminated from the population, with the remaining machines forming the observed meta-machine. Figure 8 summarizes our partitioning, and the arrow illustrates the flow of probability away from $\Sigma^{*'}$ and into machine 15. This dynamic is visible in figure 3, as machine 15 increases its presence in the population at the expense of these machines.

We next give a broad overview of the composition dynamics between these sets. Machine 15 is unique in being the only transducer with an input and output language of Σ^*, yet with a channel capacity of zero. It serves a mediator for the decay of machine's in $\Sigma^{*'}$, but is also a player in the meta-machine. Interactions involving machine 15 can lead to its self-reproduction or to the production of machines in either Σ^*_{in} or Σ^*_{out}. Compositions involving these latter sets either map back onto the set, produce machine 15, or produce machines with single arcs. Compositions involving machines with single arcs, regardless of their partner, produce machines in that set and in both Σ^*_{in} and Σ^*_{out}.

We note that in a population, equilibrium is reached through reconciliation of the forces which serve to eliminate machines from the population, and those which create them. Our transducer partitioning is a first step towards a detailed understanding of these forces, and is achieve by defining a higher level perspective on the transducers in the single state soup.

6 Conclusion

A partitioning of single state ϵ-transducers based on input and output alphabets, coupled with measures of channel capacity, provides a useful framework for understanding the population dynamics of the single state Finitary Process Soup. In particular, a transducer's channel capacity does not correlate positively, as one might expect, with its persistence in a population. The opposite is, in fact, true. The meta-machine which comes to dominate the population consists entirely of transducers with zero channel capacity. It would appear that this self-organization is driven by a force other than information flow, and thus the resulting autocatalytic network is not quantifiable in terms of channel capacity.

The composition operation may be analyzed in terms of channel capacity. Composition never increases channel capacity. In some cases it preserves capacity, in some cases it reduces it.

It should be noted that these results apply to the single state soup. Future work will seek to extend these results to understand the dynamics of multi-state soups. Transducers with more than one state correspond to channels with memory. This memory introduces a wealth of new theoretical and practical issues. A property of multi-state transducers which makes them of considerable interest is the fact that, unlike single state machines, they have a positive structural complexity, C_μ. One wonders how channel capacity varies with structural complexity, and whether the key results reported above extend to the multi-state case.

One is tempted to ask how we can hope to understand information flow in a network of machines with no channel capacity? Perhaps the answer will come by considering what is flowing on these networks. It is not necessarily codes which we are trying to communicate, it is symbols (in this case bits). Thus, communication channels may not prove to be the best metaphor. We would like to capture the fact that a simple transducer like machine 1 is capable of transmitting the symbol 0 without interference, while machine 15 completely obfuscates its input. Electrical circuits as a metaphor may be a fruitful approach.

References

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

[2] J. P. Crutchfield and O. Gornerup. Objects that make objects: The population dynamics of structural complexity. *Journal of the Royal Society Interface*, 3:345–349, 2006. doi: 10.1098/rsif.2006.0114. URL `http://rsif.royalsocietypublishing.org/content/3/7/345.full`.

[3] C. R. Shalizi and J. P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of Statistical Physics*, 104:817–879, 2001.