

# Metric Spaces for Regular Languages

Anton Xue

## 1 Introduction

Filler introduction text

## 2 Background

Filler text for background

### 2.1 Regular Languages and Finite Automata

Language recognition is a fundamental problem in theoretical computer science. Given an alphabet of unique symbols  $\Sigma$ , let  $\Sigma^*$  denote the set of all possible finite strings over the alphabet  $\Sigma$ . For some set of strings that we call a language  $L \subseteq \Sigma^*$  and string  $w \in \Sigma^*$ , we then ask if  $w \in L$ . This is the language recognition problem.

A number of problems in theoretical computer science can be formulated in terms of language recognition. Does this string belong to the set (language) of valid email addresses? Does this string belong to the set (language) of valid computer programs written in my favorite programming language? Does this string belong to the set (language) of solutions to an instance of the boolean satisfiability problem?

Here we are primarily concerned with recognizing regular languages. These are the languages that can be described by regular expressions, and see widespread application in text parsing. Equivalently stated, regular languages are precisely the set of languages recognized by the set of non-deterministic finite automata (NFA), which we aim to study here.

Formally, a NFA is a tuple  $(\Sigma, Q, \delta, S, F)$  that represents a finite state transition machine which accepts or rejects strings. Here  $\Sigma$  is the alphabet,  $Q$  is the set of states,  $\Delta : \Sigma \times Q \rightarrow 2^Q$  is the transition function,  $S \subseteq Q$  is the set of initial states, and  $F \subseteq Q$  is the set of final states.

A NFA accepts a string if there exists a sequence of transition starting from some  $q_s \in S$  that ends in  $q_f \in F$ . As the transition function maps to a set of possible states that may be arbitrarily chosen, only the existence of a transition sequence is necessary, hence the term non-deterministic.

**Example 1 (NFA).** *AAAAAAAAAAAAA*

**ALGEBRAIC FORMULATION?**

## 2.2 Metric Spaces

The concept of a distance is formalized in mathematics through a metric space. A metric space is a pair  $(M, d)$  where  $M$  is a set and  $d : M \times M \rightarrow \mathbb{R}$  is known as the metric, or distance, function that aims to assign a distance between any two members of  $M$ . A metric space comes equipped with the following axioms that must hold for any  $x, y, z \in M$ :

- (1) Non-negativity of  $d$ :  $d(x, y) \geq 0$ .
- (2) Identity of indiscernibles:  $d(x, y) = 0$  if and only if  $x = y$ .
- (3) Symmetry:  $d(x, y) = d(y, x)$ .
- (4) Triangle inequality:  $d(x, z) \leq d(x, y) + d(y, z)$ .

**Example 2.** *EXAMPLE OF METRIC SPACE HERE*

## 2.3 Measure Theory

**Filler text on measure theory**

**Example 3.** *EXAMPLE 1 of MEASURE SPACE*

**Example 4.** *EXAMPLE 2 of MEASURE SPACE*

## 3 Measure Theoretic Approaches

**MOTIVATIONAL TEXT HERE**

Consider a language  $L \subseteq \Sigma^*$ . The  $n$ -splice of a language written as  $L^n$  is defined as:

$$L^n = L \cap \Sigma^n$$

We then have the following relations:

$$L = \bigcup_{n=0}^{\infty} L^n = \bigcup_{n=0}^{\infty} (L \cap \Sigma^n) = L \cap \bigcup_{n=0}^{\infty} \Sigma^n = L \cap \Sigma^* = L$$

Suppose that  $(\mathbb{N}, 2^{\mathbb{N}}, \eta)$  is a probability measure space on  $\mathbb{N}$  with the probability measure  $\eta$ , one way to define a measure  $\lambda_\eta$  is as follows:

$$\lambda_\eta(L) = \sum_{n=0}^{\infty} \frac{|L^n|}{|\Sigma^n|} \eta(n)$$

**Theorem 1.**  $(\Sigma^*, 2^{\Sigma^*}, \lambda_\eta)$  is a measure space.

*Proof.* Since the  $2^{\Sigma^*}$  is the largest  $\sigma$ -algebra on  $\Sigma^*$ , it suffices to show that  $\lambda_\eta$  is a measure.

To see that  $\emptyset$  is mapped to 0:

$$\lambda_\eta(\emptyset) = \sum_{n=0}^{\infty} \frac{|\emptyset|}{|\Sigma^n|} \eta(n) = \sum_{n=0}^{\infty} 0 = 0$$

Now take  $(A_n) \subseteq \Sigma^*$  to be a countable collection of disjoint sets. Write  $A_n^k$  to denote the  $k$  splice of the  $n$ th set. In other words:

$$A_n = \bigcup_{k=0}^{\infty} A_n^k$$

Observe that all such  $A_n^k$  are pairwise disjoint by construction, and so:

$$\lambda_\eta\left(\bigcup_{n=0}^{\infty} A_n\right) = \lambda_\eta\left(\bigcup_{n=0}^{\infty} \bigcup_{k=0}^{\infty} A_n^k\right) = \sum_{n=0}^{\infty} \lambda_\eta\left(\bigcup_{k=0}^{\infty} A_n^k\right) = \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} \frac{|A_n^k|}{|\Sigma^n|} \eta(k) = \sum_{n=0}^{\infty} \lambda_\eta(n)$$

We conclude that  $(\Sigma^*, 2^{\Sigma^*}, \lambda_\eta)$  forms a measure space.

□

We can generalize this more. Suppose that  $\nu = (\nu_n)$  is a countable collection of measures where each  $\nu_n$  is defined on the splice  $\Sigma^n$ . Then we can extend a definition of  $\lambda_{\eta, \nu}$  as:

$$\lambda_{\eta, \nu}(A) = \sum_{n=0}^{\infty} \nu(A^n) \eta(n)$$

**Theorem 2.**  $(\Sigma^*, 2^{\Sigma^*}, \lambda_{\eta, \nu})$  is a measure space.

*Proof.* As with before, we only show that  $\lambda_{\eta, \nu}$  is a measure.

For  $\emptyset$  we have again:

$$\lambda_{\eta, \nu}(\emptyset) = \sum_{n=0}^{\infty} 0 = 0$$

Again take  $(A_n) \subseteq \Sigma^*$  to be a countable disjoint collection of sets, and  $A_n^k$  to be the  $k$  splice of  $A_n$ . Then:

$$\lambda_{\eta, \nu}\left(\bigcup_{n=0}^{\infty} A_n\right) = \lambda_{\eta, \nu}\left(\bigcup_{n=0}^{\infty} \bigcup_{k=0}^{\infty} A_n^k\right) = \sum_{n=0}^{\infty} \lambda_{\eta, \nu}\left(\bigcup_{k=0}^{\infty} A_n^k\right) = \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} \nu_k(A_n^k) \eta(k) = \sum_{n=0}^{\infty} \lambda_{\eta, \nu}(A_n)$$

This shows that  $(\Sigma^*, 2^{\Sigma^*}, \lambda_{\eta, \nu})$  is a measure space.

□

## 4 Linear Operators

### MOTIVATIONAL TEXT

For a NFA  $A$ , with  $A = (\Sigma, Q, \delta, S, F)$ , the goal is to view  $A$  acting as a linear operator between spaces. In particular, the approach we take is to see  $\delta$  as a linear operator between spaces: this is perhaps the most obvious approach, because to begin with,  $\delta$  is already the transition function.

But the questions are then: what are the appropriate linear spaces, and what are the actions of the linear operator? One initial thought is that the transition function can, in some way, be seen as a directed acyclic graph on the states  $Q$ , where each edge is weighted by the letters in the alphabet  $\Sigma$  that induce the transition. The representation as an adjacency matrix does appear in literature [1]. Roughly, if  $M$  is the transition matrix, then  $M_{i,j} \in 2^\Sigma$  denotes the states that will transition state  $q_i$  to  $q_j$ .

While such a matrix representation is useful, it is not immediately obvious how such a matrix does indeed correspond to a linear operator, especially on what linear spaces. One possible interpretation is to see the linear spaces as  $|Q|$ -dimensional, where each dimension of the space corresponds to one member of  $Q$ . The objects of the space is then sets of strings over  $\Sigma$ .

**Definition 1** (String Space). *The string space of  $\Sigma$  is a semiring  $(2^{\Sigma^*}, \cup, \cdot, \mathbf{0}, \mathbf{1})$  such that:*

- (a) *The semiring addition is the set union  $\cup : 2^{\Sigma^*} \times 2^{\Sigma^*} \rightarrow 2^{\Sigma^*}$ .*
- (b) *The semiring multiplication is the string concatenation  $\cdot : 2^{\Sigma^*} \times 2^{\Sigma^*} \rightarrow 2^{\Sigma^*}$  such that:*

$$A \cdot B = \{a \cdot b : a \in A, b \in B\}$$

Here the string space is the power set of all strings generated by  $\Sigma$  through monoid multiplication (string concatenation). Defining the string space like this allows us to equip it with a semiring structure by viewing addition as set union. For convenience, we may write  $R$  instead of  $2^{\Sigma^*}$  to denote the set corresponding to the string space.

Observe that  $R$  has the structure of a 1-dimensional linear space. The big difference, however, is that the scalar elements are elements of a semiring rather than a field. Nevertheless,  $R$  is still closed under linear operations, and is therefore a linear space.

**Theorem 3.** *A string space  $R$  is a linear space.*

*Proof.* **descriptive text here**

□

The natural extension of a 1-dimensional linear space is a  $n$ -dimensional linear space.

**Definition 2** ( $n$ -String Space). *For a string space  $R$  and  $n \in \mathbb{Z}^+$ , the  $n$ -dimensional string space  $R^n$  is then the free semimodule isomorphic to  $n$  copies of  $R$ .*

A natural representation of  $R^n$  is as a  $n$ -dimensional vector, and in this case we prefer row vectors to column vectors. We abuse notation to identify elements of  $R^n$  with their row vector representation. Furthermore, we demonstrate that this is indeed still a linear space.

**Theorem 4.** *A string space  $R^n$  is a linear space.*

*Proof.* **descriptive text** □

In particular, it would be nice to have a linear operator between string spaces.

**Definition 3** (Linear String Space Operator). *A linear string space operator is a linear operator  $A : R^n \rightarrow R^m$ .*

In particular, we are interested in a matrix representation.

**Definition 4** (Matrix Representation of Linear String Space Operator). *The matrix representation of a linear string space operator  $A : R^n \rightarrow R^m$  is a matrix  $A \in M_{n \times m}(R)$  that acts on row vectors of  $R^n$  by right multiplication.*

Here each entry of the matrix denotes the sets strings that are concatenated during transition. As with the  $n$ -dimensional string space  $R^n$ , we abuse notation for  $A$  to stand in for both the linear operator and its matrix representation. We now show that this is indeed a linear operator.

**Theorem 5.** *The matrix representation of  $A : R^n \rightarrow R^m$  is a linear operator.*

*Proof.* **descriptive text** □

Observe that the elements for the matrix of  $A$  are drawn from  $R$  (which is just  $2^{\Sigma^*}$ ) rather than  $2^\Sigma$ , which is what we would expect for an NFA. In other words, transitions in  $A$  are given by sets of potentially long strings, rather than just single alphabets. The definition provided here is intended to be slightly more general, with the NFA case of single-letter transitions being a special case. Nevertheless, given a NFA, we are now ready to describe a particular matrix representation as a linear operator between  $n$ -string spaces.

**Definition 5** (Matrix Representation of  $\delta$ ). *For a NFA  $(\Sigma, Q, \delta, S, F)$  with string space  $R$  generated by  $\Sigma$  and  $n = |Q|$ , the matrix representation of  $\delta$  as a linear operator is a matrix  $A \in M_{n \times n}(R)$  where:*

$$A_{i,j} = \{a : ((a, q_i), B) \in \delta, q_j \in B\}$$

Of course, this is a linear string space operator simply because every entry of the transition matrix will be a set of singleton strings.

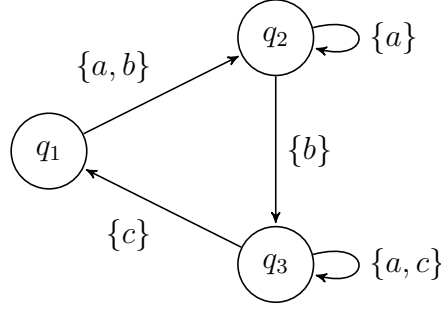


Figure 1: NFA Graph

## 4.1 Examples

**Example 5.** Consider  $\Sigma = \{a, b, c\}$  and the NFA below:

The most important distinction between this and a typical NFA is that we implicitly assume every state to be both starting and accepting. In other words, accepting strings is very permissive, which simplifies things for now. However, to embed this into our model there are several steps.

First, we have the string space  $(2^{\Sigma^*}, \cup, \cdot, \mathbf{0}, \mathbf{1})$ .

Next, for the matrix  $A$  that we will construct, take  $A_{i,j}$  to denote the transition from state  $q_i$  to state  $q_j$ . The matrix is then:

$$A = \begin{bmatrix} \mathbf{0} & \{a, b\} & \mathbf{0} \\ \mathbf{0} & \{a\} & \{b\} \\ \{c\} & \mathbf{0} & \{a, c\} \end{bmatrix}$$

In order to perform string concatenation towards the right, transition matrices act by right-matrix multiplication. That is, if  $v \in R^3$  is the initial  $n$ -dimensional string space, then the subsequent string space is  $vA$ .

To briefly demonstrate, two transitions of the matrix  $A$  appears as follows:

$$A^2 = \begin{bmatrix} \mathbf{0} & \{a, b\} & \mathbf{0} \\ \mathbf{0} & \{a\} & \{b\} \\ \{c\} & \mathbf{0} & \{a, c\} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \{a, b\} & \mathbf{0} \\ \mathbf{0} & \{a\} & \{b\} \\ \{c\} & \mathbf{0} & \{a, c\} \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \{aa, ba\} & \{ab, bb\} \\ \{bc\} & \{aa\} & \{ab, ba, bc\} \\ \{ac, cc\} & \{ca, cb\} & \{aa, ac, ca, cc\} \end{bmatrix}$$

In general, from graph theory,  $A^k$  denotes the  $k$ th consecutive transition using  $A$ , and each entry  $A_{i,j}^k$  is the set of strings that will get from state  $q_i$  to  $q_j$  in  $k$  steps.

## 5 Separating Automata

**MOTIVATIONAL TEXT**

**DISTANCE METRIC**

**Question 1.** *Given a positive set  $P \subseteq \Sigma^*$  and negative set  $N \subseteq \Sigma^*$  with  $P$  and  $N$  disjoint, does there exist an NFA  $\mathcal{A} = (\Sigma, Q, \Delta, S, F)$  with  $|Q| = n$  such that for all  $u \in P$  in the positive set  $\mathcal{A}(u) = 1$ , but for all  $v \in N$  in the negative set  $\mathcal{A}(v) = 0$ .*

We achieve this by constructing a boolean satisfiability formula that encodes  $P$  and  $N$ , and is satisfiable if and only if such  $\mathcal{A}$  exists. There are several high-level insights that we leverage:

- (1) NFAs can be represented as directed multi-edge graphs where each edge is labeled by one letter from  $\Sigma$ . Self-loops are permitted here. In other words, let  $e_{i,j,\sigma}$  be an indicator variable encodes the indicator of a transition from state  $q_i$  to state  $q_j$  on the letter  $\sigma$ .
- (2) For each  $u \in P$ , we can create a formula that forces a sequence of edge walks resulting in a final state in  $\mathcal{A}$ . Similarly for each  $v \in N$  we can encode a sequence that will force a rejection of  $v$  in  $\mathcal{A}$ .

We first construct a formula that will force  $\mathcal{A}$  to accept a word  $w$  if and only if the formula is satisfied. Let  $y_{i,t}^w$  denote be an indicator variable to show that  $\mathcal{A}$  is at state  $q_i$  at time  $t$ , with  $1 \leq i \leq n$  and  $1 \leq t \leq |w| + 1$ . Note that since each letter of  $w$  acts as a transition, the automata will occupy  $|w| + 1$  possibly repeated states during its accepting run. Then:

$$\rho_w \equiv \bigwedge_{1 \leq t \leq |w|+1} \left[ \bigwedge_{1 \leq i,j \leq n} \neg (y_{i,t}^w \wedge y_{j,t}^w) \right]$$

Forces the automata to be in only one state at any given time  $t$  while reading  $w$ . To accompany this, let  $e_{i,j,\sigma}$  to denote that  $\mathcal{A}$  has a transition edge from  $q_i$  to  $q_j$  on letter  $\sigma$ . Similarly:

$$\pi_w \equiv \bigwedge_{1 \leq t \leq |w|} \left[ \bigvee_{1 \leq i,j \leq n} (y_{i,t}^w \wedge y_{j,t+1}^w \wedge e_{i,j,w_t}) \right]$$

Additionally, we can force boundary conditions to ensure that  $\mathcal{A}$  begins reading  $w$  on a starting state and ends on an accepting state:

$$\gamma_w \equiv \left[ \bigvee_{1 \leq i,j \leq n} (e_{i,j,w_1} \wedge s_i) \right] \vee \left[ \bigvee_{1 \leq i,j \leq n} (e_{i,j,w_{|w|}} \wedge f_j) \right]$$

Where  $s_i$  and  $f_j$  are indicator variables to express that  $q_i$  is a starting state and  $q_j$  is a final state respectively. Then finally we set:

$$\varphi_w \equiv \rho_w \wedge \pi_w \wedge \gamma_w$$

Then  $\mathcal{A}$  will only accept  $w$  if and only if  $\varphi_w$  is satisfiable. Finally:

$$\Phi_{P,N} \equiv \left[ \bigwedge_{u \in P} \varphi_u \right] \wedge \left[ \bigwedge_{v \in N} \neg \varphi_v \right]$$

By construction,  $\Phi_{P,N}$  is true if and only if  $\mathcal{A}$  accepts all  $P$  and rejects all  $N$ .

Suppose that  $\Sigma$  is known. If  $\Phi_{P,N}$  is satisfiable, then  $\mathcal{A} = (\Sigma, Q, \Delta, S, F)$  can be extracted as:

$$\begin{aligned} Q &= \{q_1, \dots, q_n\} \\ \Delta &= \{((q_i, \sigma), q_j) : e_{i,j,\sigma} = \top\} \\ S &= \{q_i : s_i = \top\} \\ F &= \{q_j : f_j = \top\} \end{aligned}$$