Metric Spaces for Regular Languages

Anton Xue

1 Introduction

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

Of the many questions that can be formulated in terms of language recognition, one of central questions pertains to recognition of regular languages. Regular languages are simple yet powerful: they are indispensible in their ability to model finite state machines, describe structural patterns in strings, and behave as a simple programming language model. Abstractly, regular languages are precisely the set of languages that can be described by the class of regular expressions. Equivalently, they are the languages that are recognized by (non)-deterministic finite automata.

However, because of the way they are defined, a "flaw" is that two regular languages L_1 and L_2 may differ on a very small number of strings, yet have very different representation sizes. That is, $L_1 \cap L_2$ may be very small, but the (minimal) regular expression that can recognize L_1 may be very bit, while the regular expression used to recognize L_2 may be very small. So how different is L_1 from L_2? In other words, how might we assign a metric between L_1 and L_2, or more generally, over the space of regular languages?

One may attempt a classification with respect to their syntactic metric. That is, a metric distance is defined with respect to a representation without direct consideration for what strings the two sets may represent. Alternatively, one may seek to determine a metric in terms of the sets of strings in L_1 and L_2, which would be a semantic metric. This metric would be independent of representation, and would be defined over the space of regular sets.

Ideally, we would like a pair of metrics on the space of regular languages, one syntactic and one semantic, that are highly correlated. That is, for

AAAAAAAAAAAAAAAAAA

In this report, we present

MORE TEXT HEREEEEE

2 Background

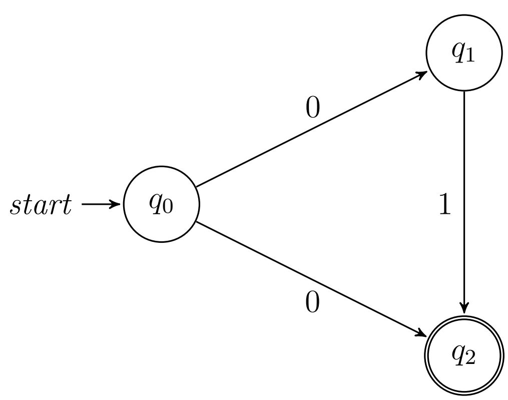
We now introduce and give a quick overview of some of the background material relevant.

2.1 Regular Languages and Finite Automata

A NFA is a tuple $(\Sigma, Q, \delta, S, F)$ that represents a finite state transition machine which accepts or rejects strings. Here Σ is the alphabet, Q is the set of states, $\Delta : \Sigma \times Q \to 2^Q$ 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)**.** *The NFA below operators over the binary alphabet* $\Sigma = \{0, 1\}$.



In order for a NFA to accept a string, there must exist a sequence of transitions (which may be non-unique). This particular NFA accepts precisely two strings:

(1) The string 0 *through the transition sequence* $q_0 q_2$.

(2) The string 01 *through the transition sequence* $q_0 q_1 q_2$.

Note that from state q_0 *there are two out-edges that are both weighted with* 0. *This is what differentiates an NFA from a deterministic finite automata (DFA). In an NFA, out-edges from the same vertex may have shared labels, but in a DFA all out-edges from the same vertex may not share labels.*

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 $d : M \times M \to \mathbb{R}$ 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 (Euclidean Metric)**.** *For some $x_i \in \mathbb{R}^n$, let x_i be the ith coordinate of the vector. Then the Euclidan (L2) metric is defined by*

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

2.3 Measure Theory

In mathematical analysis, measure theory is concerned with the rigorous formulation of "size". Such notions of size have application in generalizations of familiar concepts such as length, area, and volume, as well as integration theory. Informally, for a set X, the goal of measure theory is to assign a measure (size) to subsets of X. Often X is taken to be a space like $\mathbb{R}^n$, and common examples of subsets include intervals, rectangles, or boxes.

Formally, a measureable space is a pair $(X, \mathcal{E})$ where X is a set and $\mathcal{E} \subseteq 2^X$ is called a σ-algebra on X that satisfies the following properties:

(1) Inclusion of empty set and whole space: $\emptyset, X \in \mathcal{E}$.

(2) Closure under relative complement: $E^c \in \mathcal{E}$ if $E \in \mathcal{E}$.

(3) Closure under countable unions: $E_1, E_2, \ldots \in \mathcal{E}$ implies that

$$\bigcup_{i=1}^{\infty} E_i \in \mathcal{E}$$

The σ-algebra defined on X need not be unique. For instance, the smallest σ-algebra for any set X is $\{\emptyset, X\}$, while the largest σ-algebra is the power set 2^X.

If E belongs to the σ-algebra, that is, $E \in \mathcal{E}$, we say that E is measruable. Otherwise for some $F \in 2^X \setminus \mathcal{E}$ we say that F is un-measurable.

A measureable space can be extended into a measure space by equipping a measure $\mu : \mathcal{E} \to \mathbb{R}$ to form a triple $(X, \mathcal{E}, \mu)$. A measure satisfies the following properties:

(1) Empty set has trivial measure: $\mu(\emptyset) = 0$.

(2) Countable additivity: if $E_1, E_2, \ldots \in \mathcal{E}$ are pairwise disjoint, then

$$\mu\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} \mu(E_i)$$

Example 3 (Lebesgue Measure). *Consider the real line $\mathbb{R}$ and some interval open $(a, b) \subseteq \mathbb{R}$ with $a < b$. Intuitively one may want to assign the interval a size of equal to its length. In other words, the measure of (a, b) should be $b - a$.*

The idea of using lengths (also, area, volume, etc) as a way to measure the size of a set in $\mathbb{R}^n$ gives rise to the Lebesgue measure λ. But to formally define the Lebesgue measure, we must first define the Lebesgue outer measure $\lambda^\star : 2^{\mathbb{R}} \to \mathbb{R}$ as follows:

In order to define the Lebesgue mesure λ, we first define the Lebesgue outer measure $\lambda^\star$:

$$\lambda^\star(E) = \inf\left\{\sum_{i=1}^{\infty} I_i \ : \ \{(I_i)\} \text{ is a sequence of open intervals such that } E \subseteq \bigcup_{i=1}^{\infty} I_i\right\}$$

The difference between an outer measure and a measure is that an outer measure is defined on the largest σ-algebra, in this case $2^{\mathbb{R}}$, while a measure tends to be defined on a restricted subset.

The Lebesgue σ-algebra is a subset of $2^{\mathbb{R}}$ that is defined as the collection of all sets E such that for any $A \subseteq \mathbb{R}$ the following property holds with respect to the Lebesgue outer-measure $\lambda^\star$:

$$\lambda^\star(E) = \lambda^\star(A \cap E) + \lambda^\star(A \cap E^c)$$

This is called the Carathéodory criterion, and on such sets we set $\lambda(E) = \lambda^\star(E)$. The proof that the Lebesgue measure is indeed a measure can be found in texts on real analysis and measure theory.

Example 4 (Counting Measure). *Given a set X, the counting measure is defined on 2^X, and just counts the cardinality of each $E \subseteq X$. The counting measure tends to see application in settings dealing with finite sets.*

Example 5 (Probability Measure). *Probability measures are measures defined on the probability measure space (X, Ω, μ), where for the whole space $\mu(X) = 1$.*

2.4 Mesure Induced Metric

For a measure space $(X, \mathcal{E}, \mu)$, an interesting consequence is that a metric space can be defined on $\mathcal{E}$ as follows:

$$d(A, B) = \mu(A \triangle B)$$

Where $\triangle$ is the symmetric set difference. We now set out to show this.

Lemma 1. $(A \triangle C) \subseteq (A \triangle B) \cup (B \triangle C)$.

Proof. Observe that we may rewrite the above as follows:

$$(A \setminus C) \cup (C \setminus A) \subseteq [(A \setminus B) \cup (B \setminus C)] \cup [(B \setminus A) \cup (C \setminus B)]$$

It then suffices to show that:

$$A \setminus C \subseteq (A \setminus B) \cup (B \setminus C) \qquad C \setminus A \subseteq (B \setminus A) \cup (C \setminus B)$$

We take turns examining these.

If $x \in A \setminus C$, then this implies that $x \in A$ and $x \notin C$. There are now two cases, where $x \in B$ or $x \notin B$. First assume that $x \in B$, which will imply that $x \in B \setminus C$. Now assume that $x \notin B$, which will imply that $x \in A \setminus B$. Either way, the implication is that $x \in (A \setminus B) \cup (B \setminus C)$, and so it follows that $A \setminus C \subseteq (A \setminus B) \cup (B \setminus C)$.

If $x \in C \setminus A$, then this implies that $x \in C$ and $x \notin A$. The argument is similar to the above, in which either $x \in B$ or $x \notin B$. If $x \in B$, then $x \in B \setminus A$, and otherwise if $x \notin B$ implies that $x \in C \setminus B$. Collectively, the two imply that $C \setminus A \subseteq (B \setminus A) \cup (C \setminus B)$.

Collectively, this shows that $(A \triangle C) \subseteq (A \triangle B) \cup (B \triangle C)$. □

Theorem 1. *If $(X, \mathcal{E}, \mu)$ is a measure space, then for $d : \mathcal{E} \times \mathcal{E} \to \mathbb{R}^{\geq 0}$ defined as:*

$$d(A, B) = \mu(A \triangle B)$$

Is a metric function.

Proof. We prove the conditions necessary for a metric: identity, symmetry, and triangle inequality.

As μ is a measure, then for any $A \in \mathcal{E}$:

$$d(A, A) = \mu(A \triangle A) = \mu(\emptyset) = 0$$

By the symmetry of symmetric set difference, for any $A, B \in \mathcal{E}$:

$$d(A, B) = \mu(A \triangle B) = \mu(B \triangle A) = d(B, A)$$

For any $A, B, C \in \mathcal{E}$, we have by convexity as shown in the lemma above:

$$A \triangle C \subseteq (A \triangle B) \cup (B \triangle C)$$

Then by sub-additivity of measures:

$$d(A, C) = \mu(A \triangle C) \leq \mu((A \triangle B) \cup (B \triangle C)) \leq \mu(A \triangle B) + \mu(A \triangle C) = d(A, B) + d(B, C)$$

□

3 Measure Theoretic Approaches

Our initial efforts began from a simple question: is it possible to assign a notion of size to a regular language? We are interested in this perspective for several reasons. First, because regular languages are closed under intersection and set difference, being able to quantiatively measure their intersection and set difference would allow us to measure some type of similarity. Second, being able to embed regular languages into a measure space, as noted earlier, would allow us to derive a metric function with respect to the symmetric set difference. The natural course of investigation leads to measure theory.

Consider a language $L \subseteq \Sigma^\star$. 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:

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

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

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

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

Proof. Since the $2^{\Sigma^\star}$ is the largest σ-algebra on $\Sigma^\star$, it suffices to show that λ_η 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 $(A_n) \subseteq \Sigma^\star$ to be a countable collection of disjoint sets. Write A_n^k to denote the k splice of the nth 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 $\left(\Sigma^\star, 2^{\Sigma^\star}, \lambda_\eta\right)$ forms a measure space.

□

We can generalize this more. Suppose that $\nu = (\nu_n)$ is a countable collection of measures where each ν_n is defined on the splice Σ^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 3. *$\left(\Sigma^\star, 2^{\Sigma^\star}, \lambda_{\eta,\nu}\right)$ 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 $(A_n) \subseteq \Sigma^\star$ to be a countable disjoint collection of sets, and A_n^k to be the k splice of A_n. Then:

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

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

Having defined a measure space corresponding to the space of languages, we can leverage the earlier observation that measures induce a natural metric through their symmetric set difference.

Example 6. *Fix an alphabet Σ and a probability measure η on the non-negative integers. For the measure space $\left(\Sigma^\star, 2^{\Sigma^\star}, \lambda_\eta\right)$ and languages $L_1, L_2 \subseteq \Sigma^\star$, the distance between L_1 and L_2 can then be defined as:*

$$d(L_1, L_2) = \lambda_\eta(L_1 \triangle L_2)$$

Provided additional measures (ν_n) defined as above that extends to a measure space $\left(\Sigma^\star, 2^{\Sigma^\star}, \lambda_{\eta,\nu}\right)$, the definition then becomes:

$$d(L_1, L_2) = \lambda_{\eta,\nu}(L_1 \triangle L_2)$$

By embedding the set of strings into a measure space, we therefore effectively extend all languages over Σ into a metric space, not just regular languages. This offers a generalized framework, but comes with several issues that must be addressed before it becomes practical.

First, we did not specify the probability distributions to be used. The specific distributions of interest would most likely be problem-specific, and would require at least some justification. When specifying this metric we had in mind a goemetric distribution, which means that this metric would then favor shorter strings over longer ones. That is, two languages L_1 and L_2 would be considered more similar if they shared a large amount of shorter strings than longer ones.

Second, the summation here is infinite, meaning that a precise closed-form solution is only possible in very special cases. However because we probability distribution over the non-negative integers is used, one may leverage an approximate metric. That is, for a fixed probability distribution ν over the non-negative integers and $\varepsilon > 0$, there exists N such that:

$$\sum_{k=0}^{N} \nu(k) > 1 - \varepsilon$$

By introducing such an ε error, the sum becomes finite, and can be refined to arbitrary precision supposing that ν is computable.

4 Linear Operators

A common representation of NFAs is in terms of graphs, which in turn are often represented by adjacency matrices, adjacency lists, pointers, or some other representation. Here we are concerned with adjacency matrices. An interesting property of adjacency matrices is that they can be seen as a function that maps between vertices on a graph.

This view is interesting for several reasons. First, matrices are ways to represent linear operators between finite-dimensional vector spaces. Second, the space of linear operators can be endowed with a norm known as the operator norm defined as follows:

$$\|T\| = \inf\left\{c \in \mathbb{R} \ : \ \forall x \in X, \|Tx\| \leq c\,\|x\|_X\right\}$$

Where $T : X \to Y$ is a linear operator between normed vector spaces X and Y, which need not be finite-dimensional. The definition of a norm like this allows us to then define a metric: two operators are close in metric if their difference has a small norm. The goal is to now extend this to the space of NFAs.

There are several immediate problems to consider. For one, it's not immediately obvious (if possible) how to embed regular languages into a field, which is required for a vector space. Existing literature (much of what was also accidentally re-invented below) on the algebraic study of formal language theory is grounded in the language of (semi)-rings, which has much weaker properties than what one may desire from a field. Nevertheless, many properties from linear algebra can still be recovered, although often with a caveat. For instance, the lack of multiplicative inverses means that matrix (linear operator) inverses are hard to compute, or do not make much sense at first glance.

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 δ as a linear operator between spaces: this is perhaps the most obvious approach, because to begin with, δ 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 Σ 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 Σ.

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

(a) The semiring addition is the set union $\cup : 2^{\Sigma^\star} \times 2^{\Sigma^\star} \to 2^{\Sigma^\star}$.

(b) The semiring multiplication is the string concatenation $\cdot : 2^{\Sigma^\star} \times 2^{\Sigma^\star} \to 2^{\Sigma^\star}$ 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 Σ 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 $2^{\Sigma^\star}$ 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 4. *A string space R is a linear space.*

Proof. A semiring contains a zero element and is closed under (semiring) addition. Furthermore, it is closed under (left semiring) multiplication with respect to other elements of the semiring. □

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, where semiring operations are defined coordinate-wise. An immediate consequence is that this also forms a linear space, since it is an n-dimensional linear space.

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 $A : R^n \to R^m$.*

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 $A : R^n \to R^m$ 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. From linear algebra, we know that the space of linear operators between linear spaces is itself a linear space.

Observe that the elements for the matrix of A are drawn from R (which is just $2^{\Sigma^\star}$) rather than 2^{Σ}, 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 δ)**.** *For a NFA $(\Sigma, Q, \delta, S, F)$ with string space R generated by Σ and $n = |Q|$, the matrix representation of δ 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.

4.1 Examples

Example 7. *Consider $\Sigma = \{a, b, c\}$ and the NFA below:*

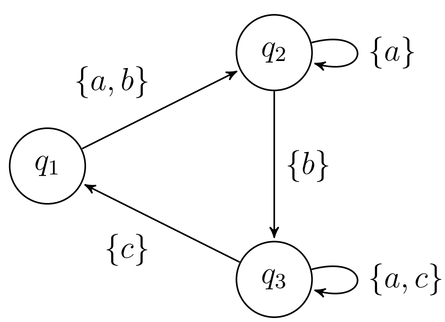


Figure 1: NFA Graph

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 $\left(2^{\Sigma^\star}, \bigcup, \cdot, \mathbf{0}, \mathbf{1}\right)$.

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 kth consecutive transition using A, and each entry $A^k_{i,j}$ is the set of strings that will get from state q_i to q_j in k steps.

5 Separating Automata

There are several ways to define a metric over the space of strings. For instance, the edit distance between two strings defines a metric.

Another metric that can be defined over two strings is with respect to an automata. For strings w and v, let $\mathcal{A}$ be the smallest automata that accepts w and rejects v. The distance can then be defined as follows:

$$d\left(w,v\right) = \frac{1}{2^{|\mathcal{A}|}}$$

Where $|A|$ here denotes the number of states of $\mathcal{A}$.

INTUITION OF WHAT THIS METIRC MEANS

The question is then how to promote this to compare over two sets of strings rather than just two strings itself. That is, given two sets of strings P and N, where we call P the set of positive strings and N the set of negative strings, let $\mathcal{A}$ be the smallest NFA that accepts all the strings in P and rejects all the strings in N, where small refers to the number of states. can a metric be defined with respect to $\mathcal{A}$? We would like to make a statement like:

$$d\left(P,N\right) = \frac{1}{2^{|\mathcal{A}|}}$$

However it's not immediately clear that this forms a metric, or if it does at all. Nevertheless, the method of finding a minimal separating NFA is still of theoretical interest.

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 Σ. 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 σ.

(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 $y^w_{i,t}$ denote be an indicator variable to show that $\mathcal{A}$ is at state q_i at time t, with $1 \le i \le n$ and $1 \le t \le |w|+1$. 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:

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

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 σ. Similarly:

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

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

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

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

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

6 Conclusion and Future Work

6.1 Graph Kernels

References

[1] John E Savage. *Models of computation*. Vol. 136. Addison-Wesley Reading, MA, 1998.