# Measures on Languages

## 1 Introduction

### 1.1 Regular Languages

**Definition 1** (Alphabet)**.** An alphabet $\Sigma$ is a countable set of distinct *characters*. □

**Example 1.** The English alphabet of 26 (lower-case) characters forms an alphabet where we have $\Sigma = \{a, b, \ldots, y, z\}$. □

**Definition 2** (Language)**.** Let $\Sigma^*$ be the set of all (possibly countably infinite) strings of a language. A language $L$ is a subset $L \subseteq \Sigma^*$. □

**Definition 3** (Regular Language)**.** A regular language $L_R$ is a language that can be *recognized* by a *deterministic finite automata* (DFA). In other words, there exists a DFA $A_{L_R}$ such that:

$$\forall s \in L_R : A_{L_R}(s) = 1$$

□

It is well known that DFAs and regular languages form a bijection. Furthermore, all DFAs can be reduced to a canonical form. If a DFA $A_{L_R}$ recognizes a regular language $L_R$, write $\eta(A_{L_R})$ to be a canonical DFA.

DFA's can also be described via *regular expressions*.

**Definition 4** (Regular Expression)**.** A regular expression $E$ is a finite string over an alphabet $\Sigma$ defined inductively, such that:

- The empty string $\varepsilon$ is a regular expression
- If $E_1$ and $E_2$ are regular expressions, then $E_1 + E_2$ is a regular expression
- If $E_1$ and $E_2$ are regular expressions, then $E_1 \cdot E_2$ is a regular expression
- If $E$ is a regular expression, then $E^*$ is a regular expression

If $E$ is a regular expression, denote the regular language generated by it as $L_R(E)$. $\square$

Regular expressions textually represent DFAs, and can be thought of as rules for how to generate a *regular language*. There should be a theorem somewhere about a bijection between regular expressions and DFAs. Here, the $+$ operator can be thought of as string choice, the $\cdot$ operator can be thought of as string concatenation, and the $*$ operator (Kleene star) denotes zero or more occurrences of a regular expression. We often elide $\cdot$ when the context is clear. The $\cdot$ syntactically binds tighter than $+$, meaning that "multiplication" distributes over "addition".

**Example 2.** Consider the alphabet $\Sigma = \{a, b, c, d\}$, with regular expression $E = ab + c + d^*$. Then the regular language $L_R(E)$ is the set:

$$L_R(E) = \{ab, c, \varepsilon, d, dd, ddd, \dots\}$$

$\square$

Finite language union, intersection, and complement with respect to a common alphabet is also well-defined for regular languages.

We can likewise also define a canonical representation for a regular expression $\eta(E)$ with the following features:

$$\begin{aligned} \eta(a) = a \qquad \eta(\varepsilon) = \varepsilon \qquad \eta(E^*) &= \eta(E)^* \\ \eta(E) = \eta(E_1) + \eta(E_2) &\iff L_R(E_1) \cap L_R(E_2) = \emptyset \\ \eta(E) = \eta(E_1) \cdot \eta(E_2) &\iff L_R(E_1) \cap L_R(E_2) = \emptyset \end{aligned}$$

We take the existence of this magical $\eta$ function for granted, because the properties that it gives for regular expression reduction are invaluable for later.

### 1.2 Measure Theory

Some talk about measure theory goes here

## 2 Pre-measures on Regular Languages

We observe a few things. First, regular languages are inherently finite in representation. Furthermore, they are also closed under finite language union, intersection, and complement. This means that they form an algebra under these operations. Therefore, it makes sense to begin by defining a pre-measure on regular languages.

## 2.1 Counting Pre-measure

Let $\mu_{0,c} \colon \Sigma^* \to [0, \infty]$ be the counting pre-measure defined for countable sets, such that:

$$\mu_{0,c}(L) = |L|$$

For a regular expression $E$, we wish to say something meaningful about the pre-measure of its generated regular language $\mu_{0,c}(L_R(E))$. Then $\mu_{0,c}$ may be defined inductively in terms of the regular expression as follows:

$$\begin{aligned}
\mu_{0,c}(\emptyset) = 0 \qquad & \forall a \in \Sigma : \mu_{0,c}(\{a\}) = 1 \\
\eta(E) = E_1 + E_2 \implies & \mu_{0,c}(L_R(E)) = \mu_{0,c}(L_R(E_1)) + \mu_{0,c}(L_R(E_2)) \\
\eta(E) = E_1 \cdot E_2 \implies & \mu_{0,c}(L_R(E)) = \mu_{0,c}(L_R(E_1)) \cdot \mu_{0,c}(L_R(E_2)) \\
\eta(E) = E_1^* \implies & \mu_{0,c}(L_R(E)) = \begin{cases} \infty & \mu_{0,c}(E_1) > 0 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

Observe that almost by definition (and a combinatorial argument), $\mu_{0,c}(L_R(E))$ counts the size of the language corresponding to regular expression $E$.

We now demonstrate that $\mu_{0,c}$ is indeed a pre-measure.

**Theorem 1.** *The function $\mu_{0,c}$ defines a pre-measure.*

*Proof.* Consider a countable alphabet $\Sigma$ and the set of all regular expressions $\mathcal{E}$.

First observe that the empty language $\emptyset$ is regular, and is generated by the empty string $\varepsilon \in \mathcal{E}$. For this we have that $\mu_{0,c}(\emptyset) = 0$, thereby satisfying the requirement that empty sets map to zero.

Now, consider two regular languages $L_R(E_1)$ and $L_R(E_2)$ that are disjoint. Recall that the semantics of $+$ means that:

$$L_R(E_1 + E_2) = L_R(E_1) \cup L_R(E_2)$$

Furthermore, because $L_R(E_1)$ and $L_R(E_2)$ are disjoint, this means that $\eta(E_1) + \eta(E_2)$ is in the desired canonical form. Hence, we expand this as follows:

$$\mu_{0,c}(L_R(E_1) \cup L_R(E_2)) = \mu_{0,c}(L_R(E_1 + E_2)) = \mu_{0,c}(L_R(E_1)) + \mu_{0,c}(L_R(E_2))$$

This satisfies the additivity rule for pre-measures. □