Mathematical Methods for Modeling Language Phonology

Zero Contradictions
ZeroContradictions.net

November 26, 2023

Contents

Introduction

Although linguistics and mathematics may be seen as unrelated fields at first glance, with linguistics typically being thought of as a social science and mathematics as a STEM field, there is actually a surprising amount of overlap between the two. This senior thesis aims to highlight some of those connections by giving an overview of the underlying mathematical structures and methods that are used to understand language phonology. For the purpose of writing a thesis paper aimed towards a mathematics-savvy audience rather than a linguistics-savvy audience, much of the terminology in this document will differ from what is normally taught and used in linguistics for the sake of easier comprehension and a better grasp of the underlying mathematics methods. Additionally, we will present examples from three different languages (Japanese, Tok Pisin, and English) in order to get a sense of how and when we may need to modify our approach for modeling a language's phonology with the greatest accuracy and efficiency.

1 Defining Sets of Sounds and Sets of Phonemes

The International Phonetic Alphabet (IPA) is an alphabetic writing system that was created in 1888 by the International Phonetic Association for precisely describing every possible sound of every spoken language in the world. New letters and diacritics have been added to the IPA every time new sounds have been discovered since its creation, and the alphabet is imperative for understanding phonetics, the study and classification of speech sounds (also known as phones).

In this section, we will use the IPA to demonstrate how sets can be used to model phonetics. We will then show examples of phonemic sound sets that exist in natural languages, as well as examples of how each allophone (element) in a phoneme (set) is actually a cognitive grouping of sounds, which will constitute our introduction to phonology, the study of how sounds systematically function in a language.

1.1 Defining Sets of Consonants and Vowels

As we can see in Appendix 1, the upper half of the IPA chart displays pulmonic consonants, non-pulmonic consonants, vowels and other pulmonic consonants that cannot be neatly displayed into the first chart. Those sounds can be modified by using the diacritics and suprasegmental markings shown in the bottom half of the chart to denote other articulatory features. We will not explain how the human vocal tract produces all the sounds since that is not the focus of this paper, but we will describe how set theory can be used to group different sounds together based on the properties that they share or don't share. Most noticeably, we can see that the pulmonic consonants are arranged with the individual rows representing the different manners of articulation, and the columns representing the different places of articulation. We can also observe that in the cells of the table that voiceless consonants are placed on left and voiced consonants are placed on the right when there are two consonants in a cell. With that being said, we can use those three properties to define dozens of different sets of pulmonic consonant sounds. Some examples include:

plosives = {p, b, t, d, ʈ, ɖ, c, ɟ, k, g, q, ɢ, ʔ}
nasals = {m, ɱ, n, ɳ, ɲ, ŋ, ɴ}
fricatives = {ɸ, β, f, v, θ, ð, s, z, ʃ, ʒ, ʂ, ʐ, ç, ʝ, x, ɣ, χ, ʁ, ħ, ʕ, h, ɦ}
alveolar_pulmonics = {t, d, n, r, ɾ, s, z, ɬ, ɮ, ɹ, l}
velar_pulmonics = {k, g, ŋ, x, ɣ, ɰ, ʟ}
approximants = {w, ʋ, ʁ, ɻ, j, ɰ}
. . .

We can can also describe subsets of those sets by combining different features, like the ones shown in the following examples:

voiceless_plosives = {p, t, ʈ, c, k, q, ʔ}
voiceless_fricatives = {ɸ, f, θ, s, ʃ, ʂ, ç, x, χ, ħ, h}
voiced_plosives = {b, d, ɖ, ɟ, g, ɢ}
voiced_fricatives = {β, v, ð, z, ʒ, ʐ, ʝ, ɣ, ʁ, ʕ, ɦ}
. . .

Additionally, linguists also define useful consonant sets based on properties that group together different rows, columns, and diacritics of the IPA chart as well as other properties that go beyond the scope of this paper, such as stops, stridents, sibilants, obstruents, consonantal sononants, implosives, coronal consonants, and many more:

stops = {p, b, t, d, ʈ, ɖ, c, ɟ, k, g, q, ɢ, ʔ, m, ɱ, n, ɳ, ɲ, ŋ, ɴ}
stridents = {f, v, s, z, ʃ, ʒ, ʧʤ}
sibilants = {s, z, ʃ, ʒ, ʧʤ}
pulmonic_obstruents = {p, b, t, d, ʈ, ɖ, c, ɟ, k, g, q, ɢ, ʔ, ʙ, r, ʀ, ɾ, ɽ, ɸ, β, f, v, θ, ð, s, z, ʃ, ʒ, ʂ, ʐ, ç, ʝ, x, ɣ, χ, ʁ, ħ, ʕ, h, ɦ, ɬ, ɮ}
consonantal_sonorants = {m, ɱ, n, ɳ, ɲ, ŋ, ɴ, w, ʋ, ʁ, ɻ, j, ɰ, ɬ, ɮ, l, ɭ, ʎ, ʟ}
implosives = { ɓ, ɗ, ʄ, ɠ, ʛ}
coronal_pulmonic_consonants = {t, d, n, r, ɾ, θ, ð, s, z, ʃ, ʒ, ʈ, ɖ, ɳ, ɽ, ʂ, ʐ, ɭ, ɻɕ, ʑ, ɬ, ɮ, ɹ, l}
lateral_consonants = {ɬ, ɮ, l, ɭ, ʎ, ʟ}
voiceless_aspirated_plosives = {pʰ, tʰ, ʈʰ, cʰ, kʰ, qʰ, ʔʰ}
palatalized_alveolar_pulmonics = {tʲ, dʲ, nʲ, rʲ, ɾʲ, sʲ, zʲ, ɬʲ, ɮʲ, ɹʲ, lʲ}
. . .

The same can be done for vowels:

close	{i, y, ɪ, ʏ, ɨ, ʉ, ʊ, ɯ, u}
close_mid	{e, ø, ɘ, ɵ, ə, ɤ, o}
open_mid	{ɛ, œ, ɜ, ɞ, ʌ, ɔ}
open	{æ, ɶ, ɐ, a, ɑ, ɒ}
front	{i, y, ɪ, ʏ, e, ø, ɛ, œ, æ, ɶ}
central	{ɨ, ʉ, ɘ, ə, ɜ, ɞ, ɐ, a}
back	{ɯ, u, ʊ, ɤ, o, ʌ, ɔ, ɑ, ɒ}
front_rounded	{y, ʏ, ø, œ, ɶ}
back_unrounded	{ɯ, ɤ, ʌ, ɒ}
long_close_mid	{eː, øː, ɘː, ɵː, əː, ɤː, oː}
open_nasalized	{æ̃, ɶ̃, ɐ̃, ã, ɑ̃, ɒ̃}
primary_cardinal	{i, e, ɛ, a, ɒ, ɔ, o, u}
secondary_cardinal	{y, ø, œ, ɶ, ɒ, ʌ, ɤ, ɯ, ɨ, ɔ}
high_tone_primary_cardinal	{˥i, ˥e, ˥ɛ, ˥a, ˥ɒ, ˥ɔ, ˥o, ˥u}
low_tone_primary_cardinal	{˩i, ˩e, ˩ɛ, ˩a, ˩ɒ, ˩ɔ, ˩o, ˩u}
rising_tone_primary_cardinal	{˩˥i, ˩˥e, ˩˥ɛ, ˩˥a, ˩˥ɒ, ˩˥ɔ, ˩˥o, ˩˥u}
falling_tone_primary_cardinal	{˥˩i, ˥˩e, ˥˩ɛ, ˥˩a, ˥˩ɒ, ˥˩ɔ, ˥˩o, ˥˩u}
. . .	

We will discuss in sections 3 and 4 why it is useful to define sets of sounds, but for now, we will turn our attention to set operations and phonemes.

1.2 Set Operations on Sets of Sounds

As we shall demonstrate, all the operations, notations, and relations commonly associated with set theory can be used to represent phonetic sound sets. The complement and power set operations could be performed on sets of sounds too, but in practice they aren't very useful, so they are not listed here.

Set Operation	Example
In	k,t,b ∈ plosives
Not In	s,w,l ∉ plosives
Subset	sibilants ⊆ stridents
Union	plosives = voiced_plosives ∪ voiceless_plosives
Intersection	voiced_plosives = plosives ∩ voiced_C
Set Difference	stridents - sibilants = {f, v}
Empty Set	∅ = plosives ∩ fricatives
Cardinality	|sibilants| = 6

In the next sections, we will use these set operations as needed.

1.3 Defining Individual Phonemes as Sets of Allophones

Phonology is the study of how languages systematically organize their sounds in the minds of their speakers. One of the primary focuses of phonology is the study of phonemes, which are cognitive abstractions (modeled as sets) of sounds, which are known as allophones. That is to say that allophones are the individual elements of a phoneme. While native speakers can consciously distinguish phonemes from other phonemes, allophones of the same phoneme do not create any meaningful changes in the words that they appear in. This is precisely the justification for modeling phonemes as sets of allophones. Every oral language has phonemes and phonemes can differ wildly from language to language, which is a key reason why different languages and different dialects sound different from each other. We will begin to see in section 3 how the various elements of a phoneme are realized in spoken speech.

In Linguistics, phonemes are identified by their underlying representations and written between /slanted brackets/, whereas allophones are written in [square brackets]. Thus, sounds written between slanted brackets are phonemes (sets), and sounds written between square brackets are elements of phonemes. The underlying representation of a phoneme is theorized to be the default allophone of that phoneme before any phonological rules or phonological constraints are applied to it (see sections 4 and 5). For that reason, the underlying representation is typically the first allophone to be listed in a phoneme's set.

Lastly, the lists of phonemes defined for Japanese and English in the following two subsections are *not* exhaustive lists of every single phoneme in Japanese and English. A complete description of Japanese and English phonemes that lists every possible allophone for every existing phoneme in those languages would be so long that it would go beyond the scope of this paper. Likewise, it should be also noted that the allophones of a phoneme can vary depending on the dialect of a language. The following subsections will give some examples of how different definitions for phonemes can be part of dialectal differences of a language, but they will again only be showing a sampling of what is possible.

1.3.1 Defining Phonemes in Japanese

Standard Japanese has the following notable phonemes. The underlying representation allophone is listed as the first element in each of the phonemes below, except for /Q/, which does not have an underlying representation.

Phoneme	Allophones
/t/	{t, ʧ, ʦ}
/s/	{s, ʃ}
/h/	{h, ç, ɸ}
/N/	{m, n, ɲ, ŋ}
/Q/	{p, t, ʧ, ʦ, k, s, ʃ, h, ç, ɸ, m, n}

Depending on the dialect of Japanese, the allophones of those phonemes and more can vary considerably. For example, in the Tōhoku dialect [d] ∈ /t/ even though [d] ∉ /t/ in Standard Japanese. As another example, [ç] ∉ /h/ and [h] ∈ /s/ for the Kansai dialect.[1] These Japanese phonemes will be further explained in section 3.2.

1.3.2 Defining Phonemes in English

The following are allophones of a sampling of phonemes in General American English.

Phoneme	Allophones
/p/	{p, pʰ}
/t/	{t, tʰ, ʧ, ɾ, ʔ, t̪}
/k/	{k, kʰ}
/d/	{d, ɾ, ʤ, d̥, d̪]}
/l/	{l, l̥, lˠ}
/æ/	{æ, æː, æ̃ː}

Once again, constituent elements of English phonemes vary depending on the dialect in question, especially for English vowels. [ɾ] ∉ /t/ ∪ /d/ in British English even though [ɾ] ∈ /t/ ∩ /d/ in General American English. [ɾ] is an alveolar flap (also called a tap) which somewhat similar to [d], but the main difference between a flap and a plosive is that in a tap/flap, there is no buildup of air pressure behind the active articulator (the tongue in this case) and consequently no bursting release of air.[2] Likewise, [ʔ] $\notin$ /t/ for some English speakers. The articulatory motivation for [ɾ] $\in$ /t/ $\cap$ /d/ and an explanation for how English phonemes surface as allophones in oral speech will be given in section 3.3.

1.4 Considerations for Defining Phonemes in Other Languages

Tonal languages can have tonemes, which are sets of tonal variations of the default tones, analogous to the relationship between phonemes and allophones. That said, tonemes can be modeled using set theory in the same way it is done for consonant and vowel phonemes.

We have already seen that it is possible for allophones to exist in two or more phonemes's sets with the case of [ɾ] $\in$ /t/ $\cap$ /d/ in General American English. In some cases, it is also possible for phonemes to have the exact same allophones, and yet still be different since the phonemes have different underlying representations. In Spanish for example, /r/ = {r, ɾ} and /ɾ/ = {ɾ, r}. If we wanted to, we could add a subscript to the allophones belonging to each of those sets to distinguish these phoneme's sets and indicate which sets the allophones belonged to before they got processed through the phonological rules.

1.5 Summary

We have seen that sounds can be organized into sets based on their phonetic properties, and we will later see in section 3 that many of these sets of sounds correspond to sets of phonemes (known in linguistics as natural classes), which are essential to define in order to describe language phonology. Every phoneme in a language is an abstract set of allophonic sounds, and the elements of a phoneme are realized according to phonological rules, also mentioned in section 4.

2 Strings and Combinatorics of Syllables and Words

Sounds are the building blocks of syllables and words, which can be mathematically defined as strings and substrings based on the set-theoretic approach that we have developed in the previous section. But before we define strings of syllables and words, we will first state some mathematical and linguistics definitions and examine some considerations.

2.1 Definitions of Mathematical and Linguistic Concepts

We shall review some discrete mathematical concepts by giving a short glossary:[3]

- An **alphabet** is a finite set of characters denoted by Σ.
- A **character** is an element of an alphabet. In phonology, all characters are phones (sounds), which can be either phonemes or allophones depending on the type of string that contains them (more on this in section 3.1).
- A **string** s is a sequence of characters of finite length. The characters of a string will be enclosed in slanted brackets (//) if it is a phoneme string, and enclosed in square brackets ([]) if it is an allophone string.
- The **length of a string** is the number of characters inside the string. It is denoted by $|s|$.
- The **empty string** is the unique string whose length is 0, and is denoted by the symbol λ.

- A **substring** is a string formed from a consecutive subsequence of characters from a string with an equal or greater length.
- The **concatenation** of strings s and t is the unique string of length $|s| + |t|$ which has s and t as disjoint substrings and has s as an initial segment. If s and t have different alphabets, the alphabet of their concatenation is the union of the alphabet of s and the alphabet of t. Concatenation is denoted by $s.t$
- For any sets A and B (whether they be alphabets or strings in this case), we define $A \times B$, the **cartesian product** of A and B, as $A \times B = \{(a, b) : a \in A \wedge b \in B\}$.
- The **product rule** says: If A and B are alphabets, then the cardinality of the set of strings $A \times B$ is $|A \times B|$, which is the same as $|A| \times |B|$.

Furthermore, we shall define some linguistics concepts in terms of the previous mathematical concepts.

- **Phonotactics** is mathematically defined as the phonological restrictions of a language which govern how syllable substrings (sequences of phonemes) can concatenate together to form valid syllables, as well as the restrictions for how syllable strings can concatenate together to form valid words.
- An **onset** is a string of consonant sounds located at the beginning of a syllable.
- A **nucleus** is a string of sounds at the center of a syllable. Every valid syllable in a language *must* have a nucleus substring within it, and may or may not have onset and/or coda strings concatenated to the nucleus string.
- A **coda** is a string of consonant sounds at the located at the end of a syllable.
- A **rime** or **rhyme** is a string composed by concatenating the nucleus and the coda (if any) of a syllable.
- A **syllable** is string of sounds formed by concatenating an onset string, a nucleus string, and a coda string, in that order.
- A **word** is a string of sounds that can be spoken in isolation with objective or practical meaning. All word strings are formed formed by concatenating syllable strings.
- A **consonant cluster** is a string of consonants that may or may not overlap a syllable boundary.

To summarize, words are strings of sounds that are formed by concatenating syllable strings, which are formed by concatenating syllable component strings (onset, nucleus, and coda).

2.2 Considerations for Creating Words and Counting Syllables

Syllables and words are both language-dependent notions. Different languages have different rules for what can and cannot be considered a syllable or word, and this can even vary across different dialects the same language, as well as from speaker to speaker. With that being said, there is plenty of room for debate and discussion regarding how we define valid syllables and words, which often relate to recent language changes, onomatopoeias, loanwords from other languages, rebracketing around syllable boundaries, and rare words/vocabulary.

To exemplify some of these issues, we will examine English. In English, there are some sounds that are very infrequent outside interjections and onomatopoeias. /?/ (the glottal stop consonant) seems to occur as a phoneme in its own right in the interjection *uh-oh*, even though it usually only exists as an allophone in most English dialects. Likewise, /x/ often occurs at the end of the interjection *ugh*, even though /x/ is absent in most modern English dialects. Some English speakers may be more familiar with these sounds than other English speakers, especially if they know foreign languages or use loan words that contain these sounds, which raises the question if they should be considered when counting all the possible English syllables.

English also has syllables with rare onsets that only occur in words of Greek origin, such as: /sf/, /sfr/, /skl/, /sθ/, /θl/, /tm/ in the words: sphinx, sphragistics, sclera, sthenic, thlipsis, and tmesis. If a native English has never heard some or any of these rare Greek words before and thus uses them in their active vocabulary, would it still be appropriate to count syllables with these rare consonant clusters when counting English syllables? This is another issue to assess.[2]

For English, there are also the onsets: /vr/ and /vl/. /vr/ occurs in the onomatopoeia “vroom, vroom”, but occurs rarely if ever in any other English words. /vl/ used to not exist in English at all until it started occuring in names and loanwords from Eastern Europe, such as ‘Vladimir’, ‘Vlach’, or ‘Vlasic’ pickles. But in the 21st century, English now has the word ’vlog’ /vlɑg/, which originated by blending the words ‘video’ and ‘blog’. ‘vlog’ and its derived terms are probably the only existing native English words in mainstream usage to contain /vl/ as an onset to date, which strengthens the case for regarding /vl/ as a valid onset.[2]

There are also dialectical, and sociolectal, and diachronic considerations. As an example of how dialect can affect the total number of syllables, although the vowel [ɔ] is a distinct phoneme in General American English, it does not exist as a phoneme in my native dialect of English (Western American English), so I pronounce the words ’caught’, ’gnawed’, and ’stalk’ in the exact same way as I pronounce ’cot’, ’nod’, and ’stock’, such that they are homophone pairs pronounced respectively as /kɑt/, /nɑd/, and /stɑk/, so I would pronounce fewer distinct syllables than an English speakers who does not have the cot-caught merger. Thus, any effort to count syllables in a language should explictly state the specific dialect in question. It might even be practical to specify the sociolect as well. In General American English, /ɹ/ may or may not be pronounced in the coda of a syllable depending on the age, social class, and ethnicity of the speaker, as well as the social context. The last major variable is that language is always changing. The language spoken today is not the same language spoken decades or centuries into the past or future. With all that being said, we shall proceed to count the syllables in the languages that we are observing for the standard dialect, in a formal social context, as of the time of this writing.

2.3 The Combinatorics of Generating Syllables: Cartesian Products and Concatenation

The most optimal approach that we we use to generate and count syllable strings will depend on how complex the language phonology is, and what it is like. For languages with rather simple phonology like Japanese, we can generate all the possible syllable strings by computing some cartesian products and making minor adjustments between those computations, and the product rule would be used to count all the syllable generated. On the other hand, languages with more complicated phonotactics like Tok Pisin and English will require first computing the cartesian product of the syllable components, and then concatenating the syllable component strings. We will also briefly discuss some of the considerations mentioned above.

2.4 Japanese Syllables and Phonotactics

Japanese has rather simple phonotactics. In linguistic notation that is standard for describing phonotactics, a Japanese syllable can be (C)(j)V(N), where C is an optional singleton or geminate consonant, V is a short or long vowel that can optionally be preceded by /j/, and N is an optional coda that only consists of a nasal phoneme that is further described in section 3.2.[1]

Vowel length is a distinctive feature across all five vowel phonemes, with an example being: tsuki 'moon' and tsūki 'airflow'. Changing the length of a vowel within a word can change the meaning of the entire word.

Consonant length is also contrastive in Japanese. The minimal pair, kita 'came' and kitta 'cut', is an example of this, however permissible geminate (double) consonants in Native Japanese words (as opposed to borrowed foreign words) are limited to a set with a cardinality of 6 elements: native_geminate_C = {p͡p, t͡t, k͡k, s͡s, m͡m, n͡n}. Foreign words however can additionally begin with the geminate consonants in foreign_onset_C = {b͡b, d͡d, g͡g, z͡z}, which are just the geminate forms of the voiced obstruent phonemes in Japanese. The geminate consonants have a tiebar written over them is to indicate that the two consecutive letters are a single element within their respective alphabets. They won't always be written with a tiebar, but the tiebars are written here for clarity and to make a point.

Additionally, Japanese syllables can begin with geminate consonants, but Japanese words cannot begin with geminate consonants, so a syllable with a geminate consonant can never be the first syllable in a word. Now we will define the onset, nucleus, and coda alphabets.

Alphabet	Phonemes	Cardinality
singleton_C	{λ, p, b, t, d, k, g, s, z, h, m, n, r}	13
native_geminate_C	{p͡p, t͡t, k͡k, s͡s, m͡m, n͡n}	6
native_onset_C	singleton_C ∪ native_geminate_C	19
native_onset_C	{λ, p, b, t, d, k, g, s, z, h, m, n, r, p͡p, t͡t, k͡k, s͡s, m͡m, n͡n}	19
foreign_onset_C	{b͡b, d͡d, g͡g, z͡z}	4
ALL_onsets_C	native_onset_C ∪ foreign_onset_C	23
ALL_onsets_C	{λ, p, b, t, d, k, g, s, z, h, m, n, r, p͡p, t͡t, k͡k, s͡s, m͡m, n͡n, b͡b, d͡d, g͡g, z͡z}	23
vowel	{a, e, i, o, ɯ, j͡a, j͡o, j͡ɯ}	8
vowel_length	{λ, R}	2
coda	{λ, N}	2

Although λ is not a consonant phoneme in any language, $\lambda \in$ singleton_C to represent syllables that don't begin with a consonant. In the same manner, $\lambda \in$ coda to represent syllables don't end with the coda nasal, because every standard Japanese syllable either ends with the coda nasal, or it doesn't. Likewise, every syllable contains either a short a vowel or a long vowel, so $\lambda \in$ vowel_length represents a short vowel, and $R \in$ vowel_length represents a continuation of the previous vowel (a long vowel).

We will also define the string /wa/. /wa/ is a valid Japanese syllable. The only reason why /w/ is not included above is because modern standard Japanese phonotactics only allow /a/ to follow /w/, and no other vowel.

In standard Japanese, the syllables /di/, /du/, /dja/, /dju/, /djo/ are merged with /zi/, /zu/, /zja/, /zju/, /zjo/, respectively. This creates 5 duplicate syllables that are pronounced exactly the same as five other syllables.

So after we compute one of the cartesian products, we will have to define a new set that will add the syllable /wa/ as an element, and we will have to subtract the duplicate syllable strings: /di/, /du/, /dja/, /dju/, /djo/. These slight adjustments will be easy to execute.

2.5 Generating and Counting Japanese Syllables

Now that we have defined all the necessary sets, we can generate all the syllables in standard Japanese by using cartesian products, and we can use the product rule to count the number of syllables, as long as we careful about the order of operations used to do both of these.

First, we compute (native_onset_C × vowel) to generate all the native Japanese syllable strings composed only of a consonant and short vowel, but as mentioned previously, we also have to add /wa/ and subtract: /di/, /du/, /dja/, /dju/, /djo/. So we will define the set native_CV using the union and set difference operations to account for these adjustments:

$$\text{native\_CV} = ((\text{native\_onset\_C} \times \text{vowel}) \cup \{/\text{wa}/\}) - \{/\text{di}/, /\text{du}/, /\text{dja}/, /\text{dju}/, /\text{djo}/\}$$

Now we will compute two more cartesian products to account for all the syllables with long vowels and/or nasal codas:

$$\text{all\_native\_syllables} = \text{native\_CV} \times \text{vowel\_length} \times \text{coda}$$

We know that the cardinality of native_onset_C is 19 and the cardinality of vowel is 8. By the product rule, the cardinality of (native_onset_C × vowel) is thus 152 elements, and the cardinality of native_CV is 148 elements because we added one element and subtracted five elements. The cardinality of vowel_length is 2, and coda is also 2. $148 \times 2 \times 2 = 592$ native syllables in the standard dialect.

$$((((19 * 8) + 1 - 5) * 2) * 2) = 592 \text{ native syllables}$$

However, all_native_syllables only contains the native syllables, and it excludes the foreign syllables that feature the foreign geminate consonants. In order to compute all the foreign syllables, we compute the following:

$$\text{foreign\_CV} = ((\text{foreign\_onset\_C} \times \text{vowel}) - \{/\text{ddi}/, /\text{ddu}/, /\text{ddja}/, /\text{ddju}/, /\text{ddjo}/\}$$
$$\text{all\_foreign\_syllables} = \text{foreign\_CV} \times \text{vowel\_length} \times \text{coda}$$

And we use the product rule and subtraction adjustment again to count all the foreign syllables. There are 108 of them total.

$$((((4 * 8) - 5) * 2) * 2) = 108 \text{ foreign syllables}$$

Additionally, there are up to 22 single-vowel single-consonant open syllables that only marginally appear in Western loanwords. It is unknown if it is permissible to double the consonant length in those foreign syllables, but they can almost certainly receive a double vowel and/or the coda consonant. We would make similar calculations if we include them. These additional foreign syllables are made possible by using some consonant allophones in non-native environments, so there isn't any easy way to briefly and systematically list all of them than just listing all of them them. Due to their nature, some Japanese speakers might not consider them valid in their idiolects, so I haven't counted them.

We could also generate and count syllables for non-standard Japanese dialects by making different adjustments. The Kagoshima and Tōhoku dialects are particularly interesting because they are the most different from the standard dialect due to their geographical distances away from Tōkyō. The Tōhoku dialect has a four-vowel system instead of a five-vowel system (plus some very different allophones, though this doesn't affect phonotactics).[1]

In the Kagoshima dialect, /di/, /du/, /dja/, /dju/, /djo/ are clearly distinct from /zi/, /zu/, /zja/, /zju/, /zjo/, so after factoring in vowel length and the coda consonant, that adds 20 more native syllables to the dialect. The Kagoshima dialect also permits two syllabic consonants, it has the additional phonemes /k^{w}/ and /g^{w}/, and it permits a couple consonant clusters [st] and [ɸt]. Although we will not calculate all the possible syllables in the Kagoshima dialect, we can generate them using the same techniques that we used for the standard dialect.[1]

2.6 Forming Word Strings in Japanese

Computing all the possible word strings for Japanese is a far more complicated issue. Although we can concatenate syllable strings to generate word strings, even if we do create a word string, there is no guarantee that that will actually be a valid word in Japanese. So any attempt to compute all the possible word strings in Japanese would actually be calculating the *theoretically* possible number. It has also been stated that words cannot start with geminate consonants, and we would of course compute the numbers of words for different syllable lengths.

I also suspect that there are probably some undiscovered phenomena that would prevent certain syllable patterns from occurring for words with 3 or more syllables, and that research would be needed to specifically state what these phenomena are and why they occur. Nonetheless, we can estimate the theoretically possible number of one syllable and two syllables words, even if some of the generated word strings are not real words. For one-syllable words, we have computed 592 native syllables in the previous section, so there would be the same number of one-syllable words.

As for two-syllable words, there are exactly 400 singleton_C native syllables:

((((13 * 8) + 1 - 5) * 2) * 2) = 400 singleton_C native syllables

$400 \times 592 = 236,800$ two-syllable words. Of course, we have already stated that not every generated word string is an actual word, so only a fraction of the 236,800 possible two-syllables words are real words. There may not be any practical applications to figuring this out, but it might be possible to somehow use this with other knowledge to reveal something about how language works.

2.7 Tok Pisin Phonotactics

For Tok Pisin, we need to take a more complicated approach. Tok Pisin has more complicated phonotactics than Japanese, so we will need to compute many different cartesian products for all the different syllable types, and then concatenate the syllable components together. Then we sum up all the different syllables for all the different syllable types. We begin with defining some alphabets:

Alphabet	Phonemes	Cardinality
single_onset	{p, t, k, b, d, g, v, s, h, m, n, w, j, l, ɾ}	15
sC_onset	{p, t, k, m, n, w, l, ɾ}	7
s	{s}	1
ɾ	{ɾ}	1
l	{l}	1
plosives	{p, t, k, b, d, g}	6
voiceless_plosives	{p, t, k}	3
non_coronal_plosives	{p, k, b, g}	4
non_coronal_voiceless_plosives	{p, k}	2
V	{a, e, i, o, u}	5
N	{m, n, ŋ}	3
coda	{p, t, k, s, m, n, ŋ, l}	8

Tok Pisin has sCLVNC phonotactics and a total of 21 different syllable types. The phonotactics notation indicates that the most complex syllable in Tok Pisin begins with /s/, followed by a voiceless plosive, a liquid consonant (/l/ or /r/), a vowel, a nasal consonant, and voiceless plosive. Tok Pisin has final obstruent devoicing (meaning that voiced fricatives and voiced plosives cannot appear in the coda) and plosives and fricatives must agree in voicing when they appear adjacently in consonant cluster strings.[4]

2.8 Forming Syllable Strings in Tok Pisin

Fortunately, we didn't have to make too many adjustments for counting the syllables in Japanese, aside from adding a syllable and removing some duplicate syllables for the standard dialect. However, this will rarely be the case for most other languages.

Most languages will probably have several exceptions in the computations of their syllable strings for various phonological reasons. That said, the safest and most reliable way to compute syllable strings and their cardinalities is to use spaces and choices multiple times, and sum up the products.

The cartesian products that are displayed below were generated using spaces and choices. Since I know the phonotactical rules for Tok Pisin, I deduced each of the 21 possible syllable types until I found all of them, and I uniquely labeled each of them according to their phonotactical pattern. From there, I created the second column by defining the sets in the previous section and displaying the set names in this section, while keeping in mind that the number of possible strings equals the number of choices to the power of the number of spaces. The third column was created by applying the product rule to each cartesian product, and the total number of syllables for each syllable type is displayed in the fourth and last column. I calculated that Tok Pisin has a total of 2280 syllables.

Regarding the notation that I use in the first column, <s>, <r>, and <l> each represent the respective phonemes /s/, /r/, and /l/. <P> stands for plosive, and depending on the context, it can represent either: 1. plosives, 2. voiceless_plosives, 3. non_coronal_plosives, or 4. non_coronal_voiceless_plosives. <V> stands for vowel, and depending on the context, <N> stands for a nasal consonant, <C> stands for 1. single_onset, 2. sC_onset, or 3. coda.

Syllable Type	Cartesian Product of Sets	Product Rule	Cardinality
λ Onsets, λ Codas	V	5	5
C Onsets, λ Codas	single_onset x V	15 x 5	75
sCV Onsets, λ Codas	{s} x sC_onset x V	1 x 7 x 5	35
PrV Onsets, λ Codas	plosives x {r} x V	6 x 1 x 5	30
PlV Onsets, λ Codas	{p,b,k,g} x {l} x V	4 x 1 x 5	20
sPrV Onsets, λ Codas	{s} x {p,t,k} x {r} x V	1 x 3 x 1 x 5	15
sPlV Onsets, λ Codas	{s} x {p,k} x {l} x V	1 x 2 x 1 x 5	10
λ Onsets, C Codas	V x coda	5 x 8	40
C Onsets, C Codas	single_onset x V x coda	15 x 5 x 8	600
sCV Onsets, C Codas	{s} x sC_onset x V x coda	1 x 7 x 5 x 8	280
PrV Onsets, C Codas	plosives x {r} x V x coda	6 x 1 x 5 x 8	240
PlV Onsets, C Codas	{p,b,k,g} x {l} x V x coda	4 x 1 x 5 x 8	160
sPrV Onsets, C Codas	{s} x {p,t,k} x {r} x V x coda	1 x 3 x 1 x 5 x 8	120
sPlV Onsets, C Codas	{s} x {p,k} x {l} x V x coda	1 x 2 x 1 x 5 x 8	80
λ Onsets, NC Codas	V x {mp,nt,ŋk}	5 x 3	15
C Onsets, NC Codas	single_onset x V x {mp,nt,ŋk}	15 x 5 x 3	225
sCV Onsets, NC Codas	{s} x sC_onset x V x {mp,nt,ŋk}	1 x 7 x 5 x 3	105
PrV Onsets, NC Codas	plosives x {r} x V x {mp,nt,ŋk}	6 x 1 x 5 x 3	90
PlV Onsets, NC Codas	{p,b,k,g} x {l} x V x {mp,nt,ŋk}	4 x 1 x 5 x 3	60
sPrV Onsets, NC Codas	{s} x {p,t,k} x {r} x V x {mp,nt,ŋk}	1 x 3 x 1 x 5 x 3	45
sPlV Onsets, NC Codas	{s} x {p,k} x {l} x V x {mp,nt,ŋk}	1 x 2 x 1 x 5 x 3	30
Sum of Total Syllables	N/A	N/A	2280

And as always, we must keep in mind that these results will vary depending on the dialect and sociolect being spoken.

2.9 Considerations for English and Other Languages

We could compute the number of English syllables and English words strings using the same approach used for Tok Pisin, but it would be even more complicated this time. Some additional considerations are that English has syllabic consonants that can be nuclei in their own right, there are even more consonants, even more vowels, and several more restrictions on how the various syllable components can concatenate together. On top of all that, there would be all the dialectal considerations for all of the world's various English dialects. It would take several more hours to write a few more pages to solve this in addition to half of the work that I have already done for it, it honestly doesn't feel very fun to make all the calculations and make sure that everything is accurate, so I will probably never finish the task, unless there's a need to finish it, perhaps for designing speech production or speech recognition software. But the task could be done nevertheless if I pressed through it and had the motivation to do it.

Now that I think of it, all the different Japanese syllable types could be displayed in a table similar to the one used for Tok Pisin. This suggests that using spaces and choices to deduce all the possible syllable types and cartesian products of sets to be computed is a potentially universal approach for all spoken languages. For tonal languages with more complex nuclei, the same approach would still work, but we would adjust things to concatenate a tone onto the nucleus string and the tone would have its own character position and index within the string. Alternatively, we could define a character for every combination of tone and nucleus to have a single character represent a tone and a nucleus.

Perhaps in the future, someone (possibly myself?) could design a software program that could automatically count and generate syllables after specifying all the phonotactical constraints of the language, dialect, and sociolect in sufficient enough detail. This would be a worthwhile investment if there was a need to count and generate syllables for thousands of different languages, dialects, sociolects, et cetera.

3 Finite-State Transducers and String Operations

When a phonological rule detects a matching substring, it takes that substring of phonemes as its input, and it outputs a new string composed of allophones.

3.1 An Introduction to Phonology and Phonological Rules (FSTs)

Phonological rules are used to describe regular, predictable changes in a language, and they are computationally equivalent to Finite-State Transducers (FSTs), hence why they are mathematical. A finite-state transducer is a type of finite-state machine that maps from an input set of characters or strings to an output set of characters or strings. In the case of phonological rules, the input string will be a phoneme string, and the output string will be the allophone string. When a phonological rule detects a matching phonemic substring (input written in /slanted brackets/), it substitutes that substring with an allophone substring (output written in [square brackets]) containing what will be pronounced.

Although FSTs can be presented as a diagrams with states and arrows mapping everything to the right places, it is impractical in practice to create for more complex phonological rules and/or sets of phonological rules. For that reason, linguists always write phonological rules using rule notation. And although represented differently than a digraph, it is still the same mathematical object as an FST. The *Sound Pattern of English* textbook published by Chomsky and Halle in 1968 has standardize how phonological rules are conventionally written, so the SPE Rule Notation will be described in this paper.

The changes that FSTs make between the input string and output string are usually string substitution operations, but character insertion, character deletion, and more are also possible operations that can be presented in rule notation.

In practice, we have to apply dozens of phonological rules to accurately describe a language's phonology, and the ordering of those many phonological rules does matter, lest the wrong output string be generated.

3.1.1 SPE Phonological Rule Notation

In SPE Notation, the following symbols are used in combination with IPA symbols to indicate the substring environments of phonological rules:[5]

Symbol	Meaning
→	"converts to"
/	"in the substring environment where"
#	Word Boundary
$	Syllable Boundary
+	Morpheme Boundary
__	"substring goes here"
C	Consonant
V	Vowel
N	Nasal Consonant
[+feature]	Feature Present in Sound
[-feature]	Feature Absent from Sound

Recall that phonemes are abstract sets of allophones. The allophone that gets realized in pronunciation depends on the phonological environment described on the right side of the phonlogical rule after the forward slash.

3.2 Writing Phonological Rules for Japanese

We will write some notable phonological rules for Japanese, using the phoneme sets described back in section 1.3.1:

- /t/ → [ʦ] / $__ɯ
- /h/ → [ɸ] / $__ɯ
- C → [+palatal] / $_{i,j}

The first rule states that /t/ becomes the sound [t͡s] when it precedes /ɯ/ in spoken speech. Likewise, /h/ becomes [ɸ] when it precedes ɯ.

As for the last rule, it applies to entire sets of phonemes. Specifically, the third rule states that any consonant becomes palatalized (is articulated closer to the hard palate of the vocal tract) when it precedes /i/ or /j/. This rule has practical motivations since /i/ is close to the palate of the vocal tract, so this would be considered a form of phonological local assimilation, when one sound becomes more similar to a sound that is directly adjacent to it.

Since this phonological rule applies to an entire set of phonemes, it is equivalent to writing several phonological rules for individual elements all at once. For the coronal subset of consonants, the palatalization goes even further and completely moves the entire consonant to the post-alveolar place of articulation, so a sub-rule could be written:

C [+coronal] → [+palatal] / $_i,j

And this phonological rule would be equivalent to all the ones below:[1]

- /t/ → [ʧ] / __i
- /d/ → [ʤ] / __i
- /s/ → [ʃ] / __i
- /z/ → [ʒ] / __i

- /n/ → [ɲ] / __i
- /ɾ/ → [lj] / __i

/N/ and /Q/ are archiphonemes, which are special phonemes that assimilate depending on whatever consonants follow them (if any, in the case of /N/). /N/ represents the coda nasal consonants that appear in some Japanese syllables, and /Q/ is the phoneme for the geminate consonant (in this theory for describing Japanese phonology).[1]

$$/\mathrm{Q}/ \rightarrow [\mathrm{C}_1]/\$\_\mathrm{C}_1$$

This phonological rule means that /Q/ becomes whatever consonant follows it. C_1 is limited to native_geminate_C = {p͡p, t͡t, k͡k, s͡s, m͡m, n͡n} for native Japanese words, or foreign_onset_C = {b͡b, d͡d, g͡g, z͡z} for foreign loanwords.

3.3 Writing Phonological Rules for English

In General American English, the phoneme /t/ has the following rules to describe when its allophonic variants occur [2]:

- /t/ → [t ʰ] / #__V
- /t/ → [ʧ] / $__ɹ
- /t/ → [ɾ] / 'V$__V
- /t/ → [ɾ] / 'Vr$__V
- /t/ → [ɾ] / 'V$__l̩
- /t/ → [ʔ] / __n̩
- /t/ → [ʔ] / __$C

Since [t] is theorized to be the underlying representation, [t] appears everywhere in English when the above phonological rules fail to apply.

Some example words to the corresponding phonological rules.

The following are some example words with /t/ that demonstrate the various allophonic pronunciations of /t/, in the order that seven the phonological rules were written above: top, trace, meter, dirty, metal, button, football.

English also has a phonological rule that vowels are pronounced longer when they precede voiced consonants:

$$\mathrm{V} \rightarrow \mathrm{V}\!:\ /\ \_\_\$\mathrm{C}[+\mathrm{voiced}]$$

And the phonological rule for the lenition of /d/ between two vowels, when the first one is stressed, and the next one is not. This is identical to the fourth phonological rule written for /t/, but written for /d/ instead:

$$/\mathrm{d}/ \rightarrow [\text{ɾ}]\ /\ \text{'V}\$\_\_\mathrm{V}$$

For the words, <writing> and <riding>, we see that the rule for vowel lengthening necessarily must be applied before the intervocalic lenition rule for alveolar plosives. Although the two words sound almost identical to each other, if we listen closely, the diphthong (a glide between two adjacent vowels in the same syllable) in <riding> sounds slightly longer than the diphthong in <writing>. This is because /d/ $\in$ voiced_C and /t/ $\notin$ voiced_C. What happened is that the vowel lengthening rule before voiced consonants was applied to both words, and then the alveolar plosive lenition rule was applied to the alveolar plosive in both words to cause them to change to an allophone that they both happen to share in the phoneme's sets. If we applied the lenition rule before the vowel lengthening rule, then the two words would sound exactly the same since [ɾ] $\in$ voiced_C. This would then cause the vowel lengthening rule to apply to <writing> in addition to <riding>, but this is not the case, so it is noticeable that this would be the wrong way to order the two phonological rules.[2]

Phonological rules also vary depending on language and dialect, so do note that the third phonological rule written /t/ doesn't exist in British English and many other English dialects.

Although phonological rules are helpful simplified models for thinking about and understanding phonology, especially to people who have not learned the more complicated statistical methods, they are simplifications nonetheless that can obscure infrequent cases where the rules don't apply as expected for any number of reasons, so FSTs have their limitations. Statistically complex, yet more accurate models for describing phonology are explained in: "Spreading in Functional Phonology" (Boersma 1998), "Sources of Non-conformity in Phonology" (Martinez 2010), and "Maximum Entropy Phonotactics" (Hayes 2007).

4 Conclusion

In this paper, we have shown that phonology is fundamentally mathematical: set theory can model sets of sounds and phonemes, string theory can model syllables and words, and FSTs can model the pronunciation of phonemes. The latter can be further refined using advanced statistical methods that may not be taught until senior-undergraduate or graduate-level math courses.

4.1 Applications of This Paper

Understanding the mathematics of phonology could give someone learning a second language an advantage in learning how to more accurately pronounce and better listen to the foreign language, in ways more similar to a native speaker. At the same time, phonology can be applied to help people people with communication disorders, which would be the field of work for a Speech Language Pathologist.

Acquiring a methodical understanding of how languages are pronounced could help us uncover text written hundreds, or thousands of years ago, even for languages that have no modern living descendants today (in such a case, the starting puzzle pieces to work with would be the pronunciations of words borrowed into languages that were documented and/or are still alive). Understanding English phonology and its history enables us to see how Shakespeare rhymed words differently between the end of his career and how he used to rhyme them (the Great Vowel Shift was ongoing at the time).

FSTs were mentioned in this paper for their ability to model phonological processes, but they also have other uses in language phonology. Namely, FSTs can also be used for describing the modifications made for borrowing words from foreign languages, for describing permanent diachronic (historical) sound changes that happened in the past or present, and could even describe how the sounds known in one dialect would change to the sounds of another dialect. Since I know the phonologies of both English and Tok Pisin, I can often make accurate predictions for what Tok Pisin words might be, despite not being able to speak Tok Pisin, which would help me learn Tok Pisin if I was intent on learning it. I could also use FSTs to predict what Portuguese words might be if I knew the phonology of Portuguese, based on my knowledge of Spanish and some of the diachronic sound changes in the Romance languages.

Understanding language phonology is also necessary to making advances in Natural Language Processing, including for speech production software and speech recognition software. I mentioned earlier in the paper that it was rather tiresome to count and generate all the syllables in English. Someone who can understand the methods described in this paper could design software that could more quickly and more manageably do this task than it ever could be done manually. A data structure could then be invented to store information essential to producing and/or recognizing sounds, which could progress the field of Natural Language Processing. Even more generally, mathematical linguistics could hold several keys to advancing artificial intelligence.

A The International Phonetic Alphabet (IPA) Chart

THE INTERNATIONAL PHONETIC ALPHABET (revised to 2015)

CONSONANTS (PULMONIC)

© 2015 IPA

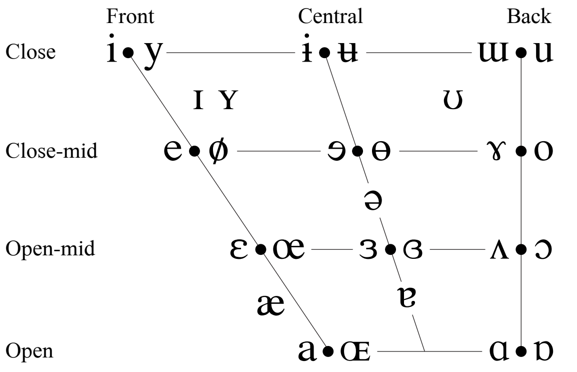
	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ	▨	ʔ ▨
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ	▨	▨
Trill	ʙ			r				▨	ʀ		▨
Tap or Flap		ⱱ		ɾ		ɽ		▨			▨
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative	▨	▨		ɬ ɮ						▨	▨
Approximant		ʋ		ɹ		ɻ	j	ɰ			▨
Lateral approximant	▨	▨		l		ɭ	ʎ	ʟ		▨	▨

Symbols to the right in a cell are voiced, to the left are voiceless. Shaded areas denote articulations judged impossible.

CONSONANTS (NON-PULMONIC)

Clicks	Voiced implosives	Ejectives
ʘ Bilabial	ɓ Bilabial	ʼ Examples:
ǀ Dental	ɗ Dental/alveolar	pʼ Bilabial
ǃ (Post)alveolar	ʄ Palatal	tʼ Dental/alveolar
ǂ Palatoalveolar	ɠ Velar	kʼ Velar
ǁ Alveolar lateral	ʛ Uvular	sʼ Alveolar fricative

VOWELS



Where symbols appear in pairs, the one to the right represents a rounded vowel.

OTHER SYMBOLS

ʍ	Voiceless labial-velar fricative
w	Voiced labial-velar approximant
ɥ	Voiced labial-palatal approximant
ʜ	Voiceless epiglottal fricative
ʢ	Voiced epiglottal fricative
ʡ	Epiglottal plosive
ɕ ʑ	Alveolo-palatal fricatives
ɺ	Voiced alveolar lateral flap
ɧ	Simultaneous ʃ and x

Affricates and double articulations can be represented by two symbols joined by a tie bar if necessary. t͜s k͡p

DIACRITICS Some diacritics may be placed above a symbol with a descender, e.g. ŋ̊

◌̥	Voiceless	n̥ d̥	◌̤	Breathy voiced	b̤ a̤	◌̪	Dental	t̪ d̪
◌̬	Voiced	s̬ t̬	◌̰	Creaky voiced	b̰ a̰	◌̺	Apical	t̺ d̺
ʰ	Aspirated	tʰ dʰ	◌̼	Linguolabial	t̼ d̼	◌̻	Laminal	t̻ d̻
◌̹	More rounded	ɔ̹	ʷ	Labialized	tʷ dʷ	◌̃	Nasalized	ẽ
◌̜	Less rounded	ɔ̜	ʲ	Palatalized	tʲ dʲ	ⁿ	Nasal release	dⁿ
◌̟	Advanced	u̟	ˠ	Velarized	tˠ dˠ	ˡ	Lateral release	dˡ
◌̠	Retracted	e̠	ˤ	Pharyngealized	tˤ dˤ	◌̚	No audible release	d̚
◌̈	Centralized	ë	◌̴	Velarized or pharyngealized	ɫ			
◌̽	Mid-centralized	e̽	◌̝	Raised	e̝ (ɹ̝ = voiced alveolar fricative)			
◌̩	Syllabic	n̩	◌̞	Lowered	e̞ (β̞ = voiced bilabial approximant)			
◌̯	Non-syllabic	e̯	◌̘	Advanced Tongue Root	e̘			
◌˞	Rhoticity	ɚ a˞	◌̙	Retracted Tongue Root	e̙			

SUPRASEGMENTALS

ˈ	Primary stress	ˌfoʊnəˈtɪʃən
ˌ	Secondary stress	
ː	Long	eː
ˑ	Half-long	eˑ
◌̆	Extra-short	ĕ
|	Minor (foot) group	
‖	Major (intonation) group	
.	Syllable break	ɹi.ækt
‿	Linking (absence of a break)	

TONES AND WORD ACCENTS

LEVEL			CONTOUR		
e̋ or	˥	Extra high	ě or	˩˥	Rising
é	˦	High	ê	˥˩	Falling
ē	˧	Mid	e᷄	˦˥	High rising
è	˨	Low	e᷅	˩˨	Low rising
ȅ	˩	Extra low	e᷈	˧˦˧	Rising-falling
ꜜ	Downstep		↗	Global rise	
ꜛ	Upstep		↘	Global fall	

Typefaces: Doulos SIL (metatext); Doulos SIL, IPA Kiel, IPA LS Uni (symbols)

References

[1] Tsutomu Akamatsu. *Japanese Phonetics: Theory and Practice*. Lincom Europa, 1997.

[2] Elizabeth C Zsiga. *The Sounds of Language: An Introduction to Phonetics and Phonology*. Wiley-Blackwell, 2013.

[3] Shyamlendu Kandar. *Introduction to Automata Theory, Formal Languages and Computation*. Pearson, 2013.

[4] Geoffrey P Smith. *Varieties of English*, volume The Pacific and Australasia, chapter Tok Pisin in Papua New Guinea: Phonology, pages 188–210. Walter de Gruyter, 2008.

[5] Noam. Chomsky and Morris Halle. *The Sound Pattern of English*. Harper & Row, 1968.

[6] Michal Temkin Martinez. *Sources of Non-conformity in Phonology: Variation and Exceptionality in Modern Hebrew Spirantization*. PhD thesis, University of Southern California, May 2010.

[7] International Phonetic Association. International phonetic alphabet chart. http://www.internationalphoneticassociation.org/content/ipa-chart, 2015. Available under a Creative Commons Attribution-Sharealike 3.0 Unported License.