SYNTAX AND FORMAL LANGUAGE THEORY

Forest Kobayashi
Department of Mathematics, Harvey Mudd College, Claremont, CA 91711

Abstract

In this paper, we will discuss the basics of Formal Language Theory (FLT), then analyze two recent papers (Jäger 2012; Fitch 2012) that discuss the connections between FLT and Artificial Grammar Learning (AGL). We will then discuss the limits of this framework as a means of studying human language acquisition, and suggest some possible directions for future work.

1. Introduction

Formal Language Theory (FLT) is a remarkably interdisciplinary subject, in that it is hugely important in at least three orthogonal academic fields: Computer Science, Mathematics, and Linguistics. In Computer Science, FLT is the foundation of Computability Theory and Complexity Theory; in Mathematics, it is fundamental to the study of Model Theory and Logic; and in Linguistics, it is an important tool for understanding structure in many aspects of natural languages, such as phonology, morphology, and syntax [LH18, Pud13, JR12].

Today, we will focus only on the applications FLT has to the field of Linguistics. In particular, we will focus on its utility in studying human language acquisition and the syntactic structure of natural languages. We begin by discussing the basics of FLT in an abstract context, and use this as a lens through which to analyze two contemporary papers ([JR12, FF12]) on Artifical Grammar Learning. Finally, we will discuss the limitations of FLT as a probe by which to measure neurolinguistic processes, and offer some suggestions for other areas of investigation.

1.1. Some Housekeeping

First, some brief notational notes. We will use $\triangle$ to denote the end of a definition and/or theorem statement, so as to make it easier for the reader to identify these visually when scanning quickly through the paper. $\forall$ and $\exists$ are read as "for all" and "there exists".

Now, some comments on style and sourcing. Unless otherwise noted, all diagrams are produced by me, and any non-canonical examples are my own. In the following sections, if no explicit citation is given for a definition/theorem/etc., it should be assumed that these definitions are based on knowledge I have acquired in my prior coursework. Finally, endnotes have been used in lieu of footnotes, and little will be lost to the reader who chooses to ignore them.

1.2. The Chomsky Hierarchy

> "There's this onion of concentric rings. If you cut this onion open... it will make you cry." - Prof. Ran on the Chomsky Hierarchy.[1]

During the later half of the 1950's, Noam Chomsky published a series of revolutionary monographs building on work done by Thue, Turing, and Post, in which he introduced the idea of a *generative grammar* (defined below), and classified families of such grammars into *a containment hierarchy*, as shown in Figure 1 [JR12, LH18]. The impact of Chomsky's work was twofold. First, it represented a philosophical shift in Linguistics away from a purely Empiricist approach to one more closely resembling the Rationalism of the modern scientific method [Hor17]. That is, it drew a distinction between *theory and form* and *data and observation*, and "[rejected] methodological dualism" [Hor17]. Second, it brought the force of exacting mathematical rigor to linguistic theory, which has allowed the field of Linguistics to benefit from theoretical advances in FLT ever since.

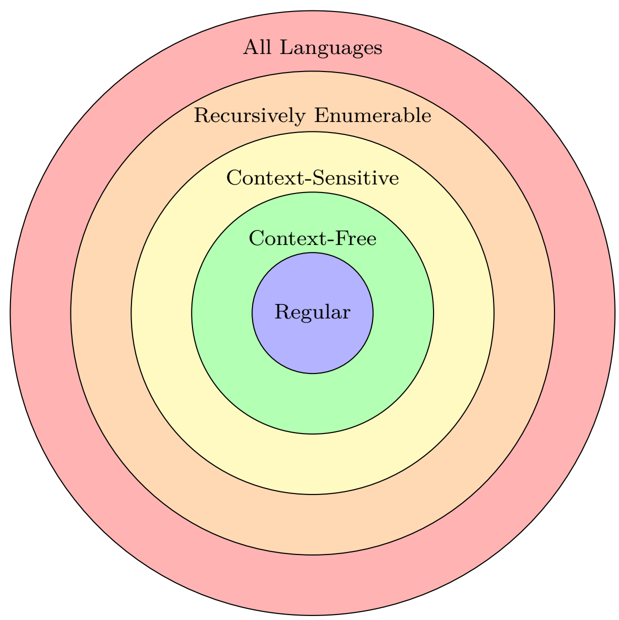

Figure 1: The Classic Chomsky Hierarchy

It is this second point that will be of particular interest to us today. As we will see in Section (2), by performing experiments on how people learn artificial grammars, we can use results from FLT to infer about the complexity of the neural architectures that handle language processing in humans. But first, we will introduce the FLT concepts that [JR12, FF12] build their arguments on. Our treatment will be more mathematically rigorous than those given in the two papers listed above, but will also be significantly terser. The experienced reader should feel free to skip this section.

DEFINITION 1.1 (Kleene Star). Let $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$ be a finite set of symbols. We refer to Σ as an *alphabet*, and the σ_i as the *characters* (sometimes called "letters") of Σ. Denote the concatenation of two characters $\sigma_i, \sigma_j \in \Sigma$ by $\sigma_i \sigma_j$. Then we define the *Kleene star* of Σ to be the set containing the empty string (denoted ε), as well as all strings formed by concatenating finite sequences of characters of Σ. Elements of $\Sigma^\star$ are referred to as *words* over Σ. △

It's worth observing that our alphabet does not have to correspond to the symbols $\{a, b, c, \ldots, z\}$. Nor do words have to correspond to what we typically think of as "words", as seen in the following example.

EXAMPLE 1.1.1. Let $\Sigma = \{0, 1\}$. Then $\Sigma^\star$ is the set of all binary strings. △

EXAMPLE 1.1.2. Let $\Sigma = \{a, b, c, \ldots, z\}$. Then up to differences in capitalization, every English word is contained in Σ. But note, not every word in Σ is a word of English.

If we were to want $\Sigma^\star$ to contain all possible sentences of English (or even, any string of sentences written in a language that uses English orthography), we could simply add the set of all punctuation symbols (including a space character) to Σ before taking $\Sigma^\star$. △

We will use the kleene star to define languages.

DEFINITION 1.2 (Language over Σ). Let Σ be an alphabet, and let L be a subset of elements of $\Sigma^\star$. Then we call L a *language* over Σ. △

The definition of language given above is general to the point of not being interesting. Hence, we want a way to define a language L over a character set Σ such that L has some amount of interesting structure. In particular, we want this structure to be *finitely describable*. This is the motivation for defining a grammar.

DEFINITION 1.3 (Grammar). Let Σ be an alphabet. Let $V = \{V_0, V_1, \ldots, V_k\}$ such that $V \cap \Sigma^\star = \varnothing$. We will refer to V as the set of *variables*. Now, suppose that we have a set of *production rules* (denoted R), each of which take the form

$$A \to B$$

Where A and B are strings of symbols taken from Σ and V, such that A contains *at least one element of* V (this requirement made so that we can't arbitrarily replace some given symbol σ_i with another symbol σ_j — we need a variable in order to be allowed to replace anything). Finally, suppose we have some variable $S \in V$ that we've selected as the "start" state (i.e., $\exists$ a rule of the form $S \to \alpha$). Then we call $\mathcal{G} = (\Sigma, V, R, S)$ a *grammar* over Σ. △

DEFINITION 1.4 (Language of a Grammar). Let $\mathcal{G} = (\Sigma, V, R, S)$ be a grammar. Then we call the *language* of $\mathcal{G}$ the set of all strings in $\Sigma^\star$ derivable in a finite sequence of steps by applying the rules of R starting on S. We denote the language of $\mathcal{G}$ by $\mathcal{L}(\mathcal{G})$. △

EXAMPLE 1.4.1 (A canonical example). Let $\mathcal{G} = (\Sigma, V, R, S)$, with $\Sigma = \{a, b\}$, $V = \{S\}$, and R by

$$S \to aSb \quad \text{and} \quad S \to \varepsilon.$$

Then $\mathcal{L}(\mathcal{G})$ is the set of all strings of the form $a^n b^n$, where a^n means concatenating n copies of a together:

$$a^n = \underbrace{aa \cdots aa}_{n \text{ times}}$$

△

The Chomsky Hierarchy classifies certain classes of grammars relative to a notion of the "complexity" of the languages they generate. We will introduce these concepts in a significantly abridged version of the treatment given in [JR12, FF12].

DEFINITION 1.5 (Regular Grammar). Let $\mathcal{G} = (\Sigma, V, R, S)$ be a grammar. Suppose that every rule of R is of the form

$$A \to \sigma \quad \text{or} \quad A \to B\sigma$$

where $A, B \in V$ and $\sigma \in \Sigma \cup \{\varepsilon\}$. Then we say $\mathcal{G}$ is *left-regular*. Similarly, if every rule of R is of the form

$$A \to \sigma \quad \text{or} \quad A \to \sigma B$$

we call $\mathcal{G}$ *right-regular*. If $\mathcal{G}$ is left-regular or right-regular, then we call $\mathcal{G}$ an *regular grammar*, and $\mathcal{L}(\mathcal{G})$ a *regular language*. △

As we can see, regular languages are heavily constrainted. For instance, the language generated in Example 1.4.1 cannot be generated by a regular grammar.

DEFINITION 1.6 (Context-Free Grammar). Let $\mathcal{G} = (\Sigma, V, R, S)$ be a grammar. Suppose that every rule of R is of the form

$$A \to B$$

where $A \in V$, and B is some string formed by concatenations of $\Sigma \cup V$. Then we call $\mathcal{G}$ a *context-free grammar*, and the language $\mathcal{L}(\mathcal{G})$ a *context-free language*.

Note, the essential part of this definition is that A is a *single* variable of V, and contains no elements of Σ. △

Context-Free grammars are significantly more powerful than regular languages. For instance, Example 1.4.1 is trivially generatable by a context-free grammar.

DEFINITION 1.7 (Context-Sensitive Grammar). Here, we'll give the definition found in [D'S16]. Let $\mathcal{G} = (\Sigma, V, R, S)$ be a grammar. Suppose that every rule of $\mathcal{G}$ is of the form

$$\alpha A \beta \to \alpha \beta,$$

where α, γ, and β are arbitrary concatenations of symbols in Σ, V, with the only requirement being that γ is nonempty. Note that α, β can be empty, however. Of course, a language generated by a context-sensitive grammar is said to be a *context-sensitive language*. △

Finally, we have the recursively enumerable languages.

DEFINITION 1.8 (Unrestricted Grammar). We follow the definition in [Gal04], modifying it to follow our conventions. Let $\mathcal{G} = (\Sigma, V, R, S)$ be a grammar. Suppose that every rule of R is of the form

$$\alpha \to \beta$$

where α, β are formed by concatenating elements of Σ and V, and α contains at least one variable in V. Then we call $\mathcal{G}$ an *unrestricted grammar*, and the language $\mathcal{L}(\mathcal{G})$ a *recursively enumerable language*. △

Unrestricted grammars are quite powerful. In fact, every possible computer program can be expressed in terms of a unrestricted grammar. Conversely, a first benchmark for any new programming language is *turing completeness* — that is, if the language were not constrained by physical limitations (e.g., finite memory, etc.), it could fully simulate any unrestricted grammar.

The Chomsky Hierarchy orders the languages generated above by containment. In a coarse sense, each tier to the hierarchy corresponds to a sense of the "complexity" of the system required to properly parse the grammar [LH18]. Additionally, one might notice that Figure 1 displays an additional level to the hierarchy — namely, the set of *all* languages. Indeed, this is intentional — there are languages that are simply not constructable using a grammar. The proof of this fact is a beautiful piece of mathematics employing Cantor diagonalization, but which we will not discuss here. Instead, we will simply summarize this in the following theorems, which we offer without proof:

THEOREM 1.1 (Containment). *Let $\subset$ denote* proper *containment. That is, if $A \subset B$, then every element of A is in B, but $\exists b \in B$ such that $b \notin A$. Let Σ be an alphabet. Then*

$$\begin{aligned} \{L \mid L \text{ is regular}\} &\subset \{L \mid L \text{ is context-free}\} \\ &\subset \{L \mid L \text{ is context-sensitive}\} \\ &\subset \{L \mid L \text{ is recursively enumerable}\} \\ &\subset \Sigma^\star. \qquad (*) \end{aligned}$$

△

THEOREM 1.2. *Each of the containments before (*) corresponds to an increasing level of complexity in the computational scheme required to simulate the grammar. However, the last containment (*) is special — the Church-Turing Thesis asserts that no "reasonable" computational model can perform arbitrary computations for the non-recursively enumerable languages. The interested reader is referred to a textbook on computability theory for further exposition.* △

The regular languages are decided by Discrete Finite Automata (DFAs), context-free languages are decided by Pushdown Automata (PDAs), context-free languages are decided by Linear Bounded Automata (LBAs), and recursively-enumreable languages are recognized by Turing Machines (TMs).[2]

Now, we'll discuss FLT's relationship to natural languages.

1.3. NATURAL LANGUAGES

At first glance, Natural Languages might seem very different from formal languages. For one thing, formal languages are just collections of symbols with no directly ascribed meaning, unlike natural languages. But we can address this problem by simply restricting our focus to syntax, disregarding pragmatics for the time being (we'll return to this later). Under this restriction, formal languages actually perform quite admirably in modeling language [JR12].

Recall the following basic rules from our syntax unit: when parsing a sentence, first begin with the start symbol S, then apply the rule S $\to$ NP VP, then NP $\to$ Det N, and so on [DoL16]. From the definitions above, it should now be clear that these correspond directly to rules in a formal grammar, where the alphabet is the set of all English words together with punctuation. We might wonder where this grammar falls on the Chomsky Hierarchy. At first, we'd be tempted to point to the existence of rules of the form S $\to$ NP VP as evidence that the language is *not* regular — after all, there are two variables on the right-hand side of the rule. But this is not necessarily true. It could be that the grammar rules given in [DoL16] are *reducible* to a regular grammar, by the addition of new variable symbols.

Thankfully, we need not wonder. In the 1950's, Noam Chomsky showed that English syntax is indeed not-regular — that is, the grammar contains rules that cannot be reduced to those of a regular language [JR12]. Thus, English requires *at least* a context-free grammar to parse properly. Similar arguments can be applied to other natural languages. In fact, "most researchers now agree that human languages require 'mildly context-sensitive' grammars" [FF12]. This is a very important result. Since DFAs (the parsing model for regular languages) cannot parse context-free grammars, let alone context-sensitive grammars, this tells us that a model for the neural architecture of the human brain must at least have some of the characteristics of a PDA. In particular, PDAs require access to some form of working memory in which results can be stored and later retrieved [JR12]. This is the focus of the two papers we'll examine today.[3]

2. Summary of Papers

We will analyze the following papers:

(a) Gerhard Jäger and James Rogers. Formal language theory: refining the Chomsky hierarchy. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 367(1598):1956, Jul 2012

(b) W. Tecumseh Fitch and Angela D. Friederici. Artificial grammar learning meets formal language theory: an overview. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 367(1598):1933, Jul 2012

In truth, much of the more challenging summary work has been done already in our discussion of the Chomsky Hierarchy (our two papers both spent ~9-10 pages detailing these results). Hence, in this section, we will focus primarily on the interesting applications they describe, namely those concerning *Artifical Grammar Learning*, which is summarized below from [FF12].

Definition 2.1 (Artifical Grammar Learning). *Artificial Grammar Learning* (abbreviated as AGL) is a family of experimental techniques for examining the mechanisms of human language acquisition, and theories of learning in general (it should be noted here that these techniques in fact can be extended to studies involving other species, but these experiments have been less standardized) . The general idea is this:

(a) An artifical grammar is constructed by choosing some alphabet Σ and some rule set R.
(b) Participants are shown valid strings generated by the grammar for brief periods of time, and asked to type them up. They are not informed that the strings follow any set pattern.
(c) Later, the participants are informed that there was an underlying pattern to the strings, and are asked to judge whether new input strings are grammatical or not.

Participants in such studies typically perform significantly better than would be expected if they were simply guessing. Furthermore, if participants are never told that there exists an underlying pattern to the stimuli, they still display preference for grammatical strings. △

We summarize some details particular to each paper as follows:

2.1. Particulars of [JR12]

The authors of [JR12] spend some considerable time working within the framework of FLT to refine the Chomsky Hierarchy, particular in the sub-regular regime (i.e., languages that do require the full power of a regular grammar to be generate). In particular, they use the formalism of automata theory to demonstrate classes of strings that could be used in AGL experiments to test parsing of sub-regular languages, and use machinery from the mathematical field of Model Theory to show that any neural architecture capable of performing such a task must necessarily be of a certain complexity. Finally, they describe methods of structuring trials such that researchers can better guarantee that the subjects being tested are learning the *correct* grammar, and not a simpler one that just resembles it.

2.2. Particulars of [FF12]

In terms of theoretical tools, the authors of [FF12] describe experimental procedures by which researchers could better control for cases when *lower-order* computational schemes can approximate higher-order ones (e.g., a DFA might be able to approximate the behavior of a PDA on a small string set with 80% accuracy). However, the bulk of the paper focuses on an analysis of results from neurological studies involving AGL. Of particular interest were the following:

(a) The role of Broca's region in processing syntactic complexity (conclusion: while the particulars of each of the leading linguistic theories explaining the observed behavior might differ, they all require that the brain be more powerful than a DFA).
(b) fMRI data suggests that Broca's region "supports the processing of structured sequences, and of supra-regular sequences in particular."
(c) Further neurological data appears to demonstrate that AGL studies most closely resemble the L_2 learning process, rather than the implicit gramaticallity judgments made by an L_1 speaker.
(d) Phylogenetic analysis shows that some of humans' close evolutionary relatives that underperform in AGL tasks have key structural differences in some portions of the brain, suggesting that said regions play an important role in processing more complex syntactic structures.
(e) Finally, they discuss probabilistic models of syntax.

We now offer some analysis, primarily on the limitations of FLT as a tool in AGL studies.

3. Analysis

We will discusses strengths of both papers together, then comment on implications to linguistic theories, and finally give some criticisms.

3.1. Strengths

Both papers do an excellent job at employing abstract mathematical machinery in an appropriate manner (this is a strength of [JR12] in particular). Theoretical results are given a degree of rigor appropriate for the scope of the papers, and in most cases, definitions are given carefully, and edge cases are appropriately addressed.

While the reader might be concerned that this not so much a comment on the strength of the authors' exposition on *linguistics* as it is might comment on their exposition on *mathematics*, we argue that this is not the case. While of course the primary focus of the papers should rest on linguistic theory, it is *imperative* when applying mathematical tools that one preserve the rigor with which they are defined.

This is because the main strength of mathematics as a tool for the natural sciences comes from its rigor. Using abstract mathematical tools, theories can be formulated *precisely*, as can their failure conditions. As such, whenever it is appropriate, presenting theories in the language of formal mathematics enables researchers to more easily translate said theories into well-designed experiments that minimize confounding factors. Again, we want to stress that this is not always the appropriate approach — but in the context of syntax and cognitive complexity, it is certainly a natural path to pursue.

A particular strength of [FF12] (which [JR12] did not pursue) is its emphasis on employing real-world data from modern brain scanning techniques to support the legitimacy of the theories presented. We will discuss this further in the comparisons section.

3.2. Impact

As we have discussed the ramifications of AGL in multiple previous parts of the paper, we will stay concise here and opt for a summary in lieu of a full analysis.

The techniques of Formal Language Theory are a particularly apt tool with which to analyze cognitive complexity of the human brain. In particular, they provide a non-invasive tool with which to study syntactic processing and L_2 acquisition, while simultaneously encoding a relatively high degree of meaningful results for theoretical linguistics. This makes AGL tools an important method for studying syntax, and hence for developing better Natural Language Processing tools in computational linguistics.

3.3. Criticisms

While FLT has certainly cemented itself as a lasting tool for studying the syntactic structure of Natural Languages, it has a crucial flaw: fundamentally, it ignores all semantic meaning contained in sentences (and fails even more drastically when applied to utterances instead). For the purposes of determining a crude lower bound on the complexity of language processing in the human brain, this is not as much of a concern. However, it might prove difficult to tighten this bound in future work without more complex models. We will describe two possible ideas for further work in this direction.

First, we should note that when parsing a sentence of a natural language using FLT, the symbols that we substitute (e.g., S $\to$ NP VP) represent *classes* of words, not words themselves. As such, it would be desirable to create experiments in which participants are trained on sentences involving *categories* of words (instead of simple A^nB^n patterns), and test gramaticallity judgments in such contexts. Of course, this would require a much longer training and testing period, but this would more closely resemble the abstract inferences that language learners must make to master speech.

A more ambitious task would be to try and incorporate considerations of semantics and pragmatics into AGL studies. Traditionally, FLT has ignored any and all semantic meaning of the sentences in question (indeed, this an axiomatic premise of FLT). However, since language is fundamentally communicative in nature, it would be desirable to create experiments in which the artificial grammars encode some degree of semantic meaning, and test subjects' performance on identifying combinations of gramaticallity *and* felicity. But first, we need to develop a formal mathematical theory akin to FLT that will aid in accomplishing this task. One tool that might offer a step in the right direction is the mathematical subject of Category Theory, together with some form of statistical model. While we will not get into details here, we will offer a sketch of why it might be useful.

Recall the following ambiguous sentence we covered in class: "The man saw the dog with binoculars". This sentence is structurally ambiguous — a priori, we do not know whether the man is using the binoculars to see the dog, or whether the man saw a dog, and in particular, he saw the dog that had binoculars. However, we know that the former is much more likely — first, binoculars are objects whose primary purpose is to be used to see things, hence we suspect "with binoculars" applies to the verb "saw", and second, it is hard to imagine a plausible scenario in which a dog would be in possession of a pair of binoculars. By contrast, if the sentence read "the man saw the birdwatcher with binoculars", things would be much more ambiguous, as birdwatchers are frequently in posession of binoculars. Hence, there is a sense in we expect "with binoculars" to only apply to certain categories of objects. While this kind of exception can be treated by simply refining the granularity of the parsing grammar one employs, it is not necessarily the most natural approach.

Perhaps a better example would be the following famous sentence "colorless green ideas dream furiously". While syntactically correct, it has no easily discernable semantic meaning. "Colorless" and "green" appear to be mutually contradictory, unless "green" is interpreted as meaning "new at", "environmentally friendly", or the like. But even then, it is unclear how any of these could be descriptors for the noun "ideas", and so on. Sentences like this indicate that *semantics* and *syntax* are not the same thing.[4]

Here, the perspective of Category Theory might be useful. Essentially, the development of Category Theory was motivated by the observation that oftentimes in mathematics, studying functions between objects is a useful way of gaining rich understandings about the structures themselves. The fundamental objects of Category Theory are Categories, which are defined by a set of *objects* (for now, think of these as *collections* of words), together with functions (also called *morphisms* or *maps*) between these objects, such that laws of *composition* are satisfied. One way this could be applied is by thinking of words in a sentence as *functions* that modify other words in the sentence.[5]

For instance, the word "green" could be thought of as an alias for many different functions (one for each semantic definition of "green"), and takes as arguments the appropriate category of words it can modify (note that this input is *particular* to the definition of "green" in use). The output, say "green car", is now an object that lives in the category of noun phrases. However, there is still ambiguity — "green car" could mean "eco-friendly car", or "car that is green in color". Thus, while parsing further through the sentence, we would look for other words modifying the phrase "green car" that require one type of argument, and not the other (e.g., "the lime green car". Here, "lime" can only be applied to "green car" if "green car" is in the category "cars that is green in color"). Using Category Theory, it might be possible to tie semantic meaning to syntactic validity in this way.

4. Comparison of Methodologies

Again, we will be brief here, as both of the papers we analyzed were largely theoretical. Whereas [JR12] focused mainly on theoretical results and methods of improving future experiments, while [FF12] presented some meta-analysis of data from neurological experiments. As such, a direct comparison of methodologies feels a little inappropriate. However, it is worth saying that this gives the discussions of cognitive complexity in [FF12] some amount of empirical legitimacy, which [JR12] lacks.

5. Discussion

In this paper we provided a rigorous introduction to formal language theory, and provided analyses of [JR12] and [FF12]. These articles dealt with the applications of formal language theory to artificial grammar learning, and showed such methods can serve as robust tools by which to measure neurological complexity in syntactic processing. In the latter portion of the paper, we discussed the potential applications of Category Theory to the problem of incorporating semantics and pragmatics into such studies, and how one might design AGL experiments to more closely resemble natural language processing.

Notes

[1]In all fairness to Prof. Ran, this was quickly followed with "No! It will make you smile! It's beautiful! It's a beautiful onion! It's a beautiful onion!"

[2]For all intents and purposes today, "recognize" and "decide" mean the same things, but as technical terms, they are not equivalent.

[3]We are glossing over some technical details about memory constraints in PDAs (in particular; PDAs have infinite access to what is known as "stack" memory). These concerns are certainly valid, and should not be discounted, however they escape the scope of our current analysis. We refer the reader to §3 of [FF12] for more details.

[4]Note, of course, that this does not necessarily mean that they are *disjoint*, just that they are not identical.

[5]This in itself is not necessarily a new idea — see the concept of a *Categorial Grammar*.

References

[DoL16] the Ohio State University Department of Linguistics. *Language files: Materials for an introduction to language & linguistics*. Columbus: Ohio State University Press., 12 edition, 2016.

[D'S16] Deepak D'Souza. Lecture Notes on Context Sensitive Grammars, Dec 2016. [Online; accessed 19. Dec. 2018].

[FF12] W. Tecumseh Fitch and Angela D. Friederici. Artificial grammar learning meets formal language theory: an overview. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 367(1598):1933, Jul 2012.

[Gal04] David Galles. Lecture Notes on Automata Theory, September 2004. [Online; accessed 19. Dec. 2018].

[Hor17] Norbert Horstein. On Syntactic Structures, Jan 2017. [Online; accessed 18. Dec. 2018].

[JR12] Gerhard Jäger and James Rogers. Formal language theory: refining the Chomsky hierarchy. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 367(1598):1956, Jul 2012.

[Lev13] Stephen C. Levinson. Recursion in Pragmatics. *Language*, 89(1):149–162, Mar 2013.

[LH18] Ran Libeskind-Hadas. Lecture notes in computability and logic, November 2018.

[Pud13] Pavel Pudlák. *Language, Logic and Computations*, pages 65–155. Springer International Publishing, Heidelberg, 2013.