Formal Language Theory in Computer Science and Linguistics

by Philippe Larkin

November 12, 2024

1 Introduction

Formal Language Theory has had a profound impact on the fields of computer science, linguistics and math. In computer science, FLT is used for understanding the theoretical side to computers, in other words the internal logic of a machine. The terminology used in this field of computer science sprung out of linguistic thought; the works of Noam Chomsky, built on earlier works by Alan Turing, Axel Thue and Emil Post (Jäger & Rogers, 2012). FLT quickly became very useful as a tool for teaching abstract computer mechanisms, such as lexical analyzers and parsers, which are elements of the compiling stage of a program (Linz & Rodger, 2022). In linguistics, Formal Language Theory was an early formalization attempt of the field of generative grammar. In the wake of the cognitive revolution, there was an increased desire to turn to the Scientific Method in fields which were typically less associated with pure science. With FLT, we can explain how languages accept certain strings that are created from grammars using a set alphabet. The result is either accepted or not by said language. Formal Language Theory and the many other works alongside it launched linguistics into its more modern form, on a path to describing natural languages and their many intricacies (in Chomsky's case with a great emphasis on Universal Grammar).

Along the way, the field that was born from this cognitive revolution has had its proponents and detractors. The interaction of linguistics with a variety of new fields, such as psychology, neuroscience and computer science, has led to great discoveries, but also misunderstandings. Methodologies and assumptions that had worked for some fields, like the use of statistical models to predict linguistic behavior, while useful in the psychological field, is not always applicable to linguistic phenomena. The increasingly popular field of generative artificial intelligence and its large language models (term used for a probabilistic computerized language model) have generated a new skepticism for the innate structures proposed by generative grammar as a model for language. This is exemplified in the paper by Steven Piantadosi, which will be discussed.

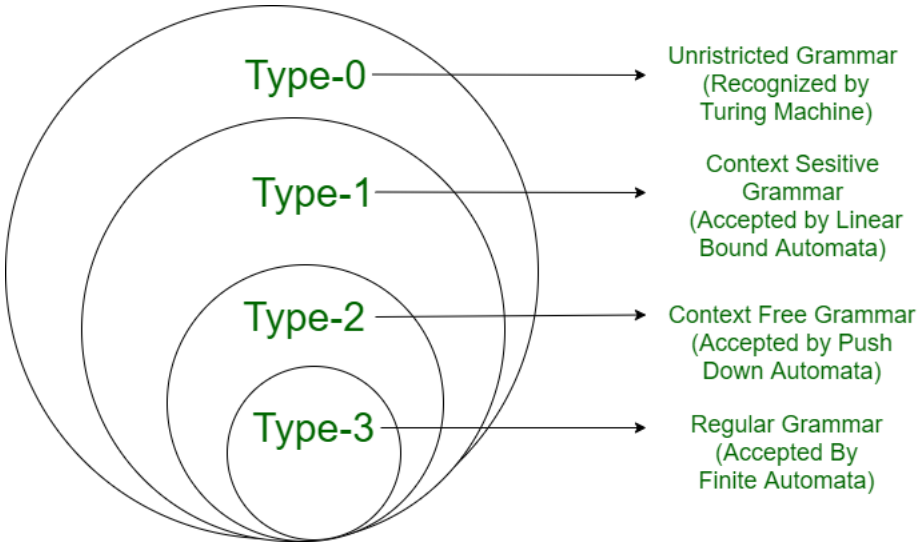
This paper will be first a small overview of Formal Language Theory in Computer Science and Linguistics, including an introduction to its concepts. Then it will touch on some differences between their approaches and how FLT is used as an important framework to this day using examples from Stuart Shieber's data on Swiss German, and an interesting proposal to integrate formal language into LLMs. A powerful parallel can be drawn between Shieber's conclusions and Chomsky's conclusions in *Syntactic Structures*, which will be further discussed. Finally, in keeping with the theme of the positives of the generative approach, I address some current criticisms of it by summarizing some arguments made against the Steven Piantadosi paper on GPT's being the end of Chomsky's approach to language, and discuss some writing on the possible overestimation of the intelligence of LLMs. Together, this paper aims to make clear the importance and relevance of

formal language theory in a mostly generative framework, while also addressing modern discussions surrounding linguistic approaches as a whole.

2 Formal Language Theory in Computer Science

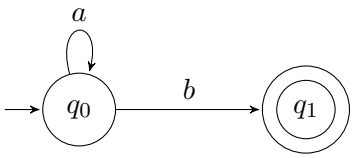
To start, using Peter Linz's *An Introduction to Formal Languages and Automata*, formal languages are an abstraction of the general characteristics of programming languages. The sentences are entities made from the symbols and rules that constitute the formal language. The formal language is the set of all sentences that are allowed by these rules. In computer science, the automaton is the abstraction of a digital computer, input, output, (sometimes) storage, and the capacity to make certain decisions which transform the input. The use of the term "formal language" refers to the set of sentences (which themselves are a combination of pre-determined symbols) which is accepted by the rules that generate the language. Thus, when we give an input, we can say whether or not this is accepted by the language (with more complexity to come) (Linz & Rodger, 2022).

The structure of many teachings in automata theory follows an ascending order of complexity of languages, called the Chomsky Hierarchy. With increasing complexity comes an increasingly powerful machine which can generate it. More powerful machines generate more complex machine because they can remember more inputs, move around back and forth on the input string, all depending on the machine. In its simplest form (refinements have been made over the years), the Chomsky Hierarchy is a nested representation of increasingly complex languages, starting with regular languages, context-free languages, context-sensitive languages, and finally recursively-enumerable languages.



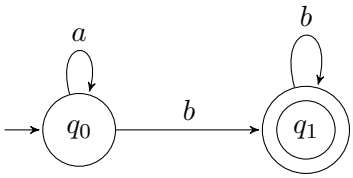
The above image (GeeksForGeeks, 2023) shows the place a regular language occupies in the hierarchy. This regular language must have a finite accepter (described by deterministic and non-deterministic automaton). For example, the language accepting $a^n b$, meaning any number of a's followed by a single b, is a regular one. The automaton we use to describe regular languages are made up of 5 parts $M = (Q, \Sigma, \delta, q_0, F)$. Q is the set of states needed to describe the language (finite), Σ is the alphabet (meaning the symbols we use), δ is the "transition function", which tells us what operations to apply to an input, q_0 is the initial state, and F is the set of accepting states (sometimes called final). So for the language described above, the non-deterministic finite accepter

would look something like:



The leftmost arrow indicates this is the start point, and is convention to include. Starting at the state q_0, the loop above indicates a loop. The loop means we can stay on q_0 and keep reading as many a's as we want. The arrow with the b above it indicates that once the amount of a's from the input string are read, if the next input is a b, the machine goes to the accepting state q_1. With nowhere to go after q_1, if anything is contained on the input string after the b, it cannot be accepted by this machine. This configuration with the loop allows us to accept the input of just b, with no a's, as an answer. If there had been a transition with a to the accepting state, we could not have accepted the empty string.

One should quickly notice the limitation of such a language. For example, a common example given is a natural evolution of the first example, such as the language $L = a^n b^n$. Some accepted inputs would be: $\lambda, ab, aabb, aaabbb$. The symbol λ (lambda) represents the empty string (for the language $a^n b^n$ this would be no a's and no b's). Since a and b share the same variable n, this means that to decide how many b's are allowed in an input, the machine has to wait and count the number of a's, count the b's and check that they are the same (it must also verify that the a's precede the b's of course). The instinct might be to draw some variation of the earlier NFA:



The problem with this is there is no limit or counting going on in a finite accepter like this. The above machine could accept inputs such as bbb which should **not** be accepted. Now, following the Chomsky Hierarchy up, we now find ourselves in need of a more "powerful" machine, one which has memory. The language $L = a^n b^n$ is a context-free language. A context-free language does have a memory component, and the important difference can be shown using what computer scientists refer to as a "grammar". Grammars are simply just another way of describing languages, and are the end product we seek, since the use of automaton is really just a way to know what kind of grammar can be associated with a language family (Linz & Rodger, 2022).

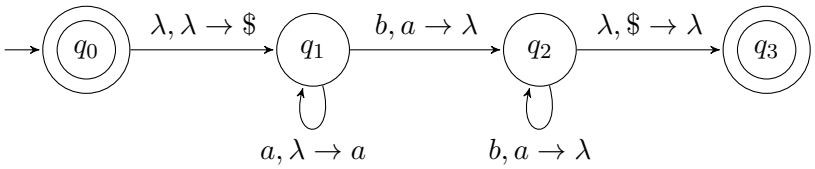
In a regular language, the left side of the grammar must only be one variable and the right side also has various limitations. We remove the right-side limitation, keep the left side to one variable, but this allows us more flexibility and can cause sort of "nested" structures to appear. In programming languages, this nesting is of course of huge importance, or else no complex program could be utilized (Linz & Rodger, 2022). $a^n b^n$ described earlier has a grammar:

$S \to aSb$
$S \to \lambda$

After reading an a, the grammar gets to an S, which forces it to go back to one of the S's, either starting a new sequence aSb, or inserting the lambda which indicates an empty string. As

the grammar forces the a and b to come in a pair, this is how an equal number of a's and b's can be realized with a grammar. It maintains equality using this rudimentary memory concept.

Push-down automaton is the term used for the machine which represents context-free languages. The "memory" component can be more easily understood using a visual. It looks something like:



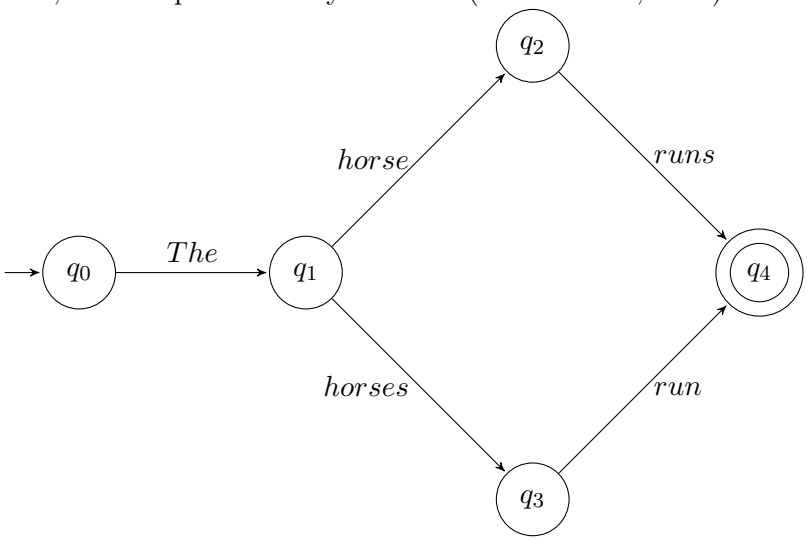
This is a non-deterministic PDA, meaning it doesn't specify where the machine goes for every single possible outcome, but nonetheless the above machine describes the language $a^n b^n$. The PDA's "memory" that we've been alluding to refers to the use of a stack. A stack is an abstract data type that collects all kinds of elements, and the order with which we add said elements is referred to as LIFO (Last In First Out). In the above machine, from left to right, the transitions can be read as "Read", "Pop", "Push". In simple terms, using the loop on q_1 as an example, if you read an "a" on your input string, you should pop nothing (λ), and push an "a" onto the stack. Pushing can be understood as adding, and popping can be seen as "removing off the top". Since as you continue pushing symbols, the older symbols on the stack get pushed lower, the most recent symbol you pushed would be the first to be popped, hence "Last In First Out". The reason this is useful is in this simple example, you can push any symbol every time you see an a, and if you see a b you pop the same symbol. If by the end of your input string the stack is empty (same number of pushes and pops), you've confirmed (with some caveats) that you have the same number of a's and b's. This is the type of basic memory that we are referring to. The \$ sign is used at the bottom of the stack by default just to indicate it is the bottom (it can be any symbol).

After the context-free languages and the PDA's, there are more levels and machines, going all the way up to Turing Machines, to represent ideally any possible computational task.

Most importantly, this application of FLT, which was co-opted from the at the time growing field of linguistics, proves incredibly powerful and useful in describing computers. Despite this, its counterpart in linguistics has gone through its own unique development in the utility of the machines, relevance, and even just overall interest to linguistic inquiry. A computer (for the foreseeable future at least), is a machine which receives an input and is instructed on what to do by a programmer. In comparison, the task of discovering what the brain is computing during any kind of real world language operation is a difficult one. What is learned, what is innate? This is at the center of a decades-long debate. The impact that FLT has had on the field of computer science is immense and is a prime example of how fundamental human inquiry can be, so far-reaching that it would involve multiple branches of science. Using FLT in computer science theory as an example, the influence of linguistics during the period marking the early cognitive revolution cannot be understated, and much of this is thanks to the advancements made specifically in generative grammar.

3 Formal Language Theory in Linguistics

Chomsky first addresses the regular grammars, or Type 3, as they were called in *Syntactic Structures*. His description works the exact same way as a finite state machine, meaning there is a set of states needed, an initial state, a set of accepting states, a transition function, and of course an alphabet which tells us which symbols we may use. The graphical representation is also exactly the same, so a simple FSG may look like (Isac & Reiss, 2013):



This simple example expresses an attempt at visualizing the structures that allow us to compute language. It generates exactly two accepted sentences, "The horse runs." and "The horses run.", but it also has a set of very glaring limitations. In *Syntactic Structures*, Chomsky contends that natural languages cannot be described with this level, as something like syntactic dependencies (if, then) do not work with FSG's. This is similar to the limitation described in section 2, as there was a missing memory component that could allow us to compute more complex languages. If speaking an accepted sentence requires mirroring a past part, or even just remembering a past part, this cannot be done by an FSG as there is no memory component. The context-free grammars are the next step up in power. These failed for their own reasons, which will be partially explored in section 4. According to Chomsky, there is a great doubt that the answers to the question of innate structures would be found only in the mathematical "Markov processes" (the term for a math model which the FSG's fall into) (Chomsky, 2002).

FSG's were not sufficient, and the conclusion was that "if we make a fsg powerful enough to not undergenerate, then it will overgenerate [...]" (Isac & Reiss, 2013). Although Finite State Grammars were never going to properly capture the complexity of natural language, this first step in attempting to put on paper some possible innate structures that allow us to do all sorts of complex things with language was an important early step, which cannot be said enough. The intuition that humans have innate capabilities is a powerful one. If the argument for the relevance of FLT was strong in relation to the field of computer science, it's even stronger in linguistics. FLT has remained of interest over the years. So long as generative grammar continues to model and predict linguistic behavior as it still does, there will be continuing motivation to solve "big picture ideas", like placing human language definitively in one of the categories of the Chomsky Hierarchy.

A Few Differences in the Approaches of Computer Science and Linguistics

Theories in computer science and theories in linguistics are, from the get go, inherently different. Linguistics attempts to understand a process that is (as far as we know) uniquely human: language. Like any field which deals with the mind, the complexity of the human brain makes it hard to really understand what is happening when a baby learns a language. Unlike genes, which we can often test for certain markers for certain diseases, the brain's intricate connections make it extremely complex. It contains, by some newer estimates close to 1 quadrillion synapses, or connections (Kemmerer, 2015) with our galaxy containing 100 billion stars for scale (NASA, 2015). Within those connections lie the secrets to language, and at the moment we do not have the tools to decipher those secrets. In the meantime, time is more usefully spent using the data available to us to determine how the linguistic system may work. Although recent work in other related fields like neuroscience and psychology is often contradictory with assumptions made in linguistic inquiry, it is not so impossible to think that language may be one of many things our very complex brain is wired to do naturally.

Considering the fact that computers are human-made creations, the way FLT is utilized in computer science is quite different. As previously mentioned, FLT is good at abstracting computer processes, at describing the grammar of a computer on a more logic and math-oriented level, rather than the engineering side. FLT is in essence describing the product of rules and restrictions of our own doing. This is already a big departure from linguistics on a fundamental level. In linguistics, rather than describing the product of a machine containing rules of our choosing, FLT seeks to establish a possible mechanism, a falsifiable proposition on the language capacity of humans. It can describe all manners of outputs and restrictions, which can be tested. All layers of the Chomsky Hierarchy, for example, come with their own set of rules, and when exceptions are found in natural language the theory can shift. From similar beginnings, FLT in computer science and in linguistics is different on many levels. However, the formalization attempts in linguistics started in the cognitive revolution have wedged linguistics into the overall more "scientific" cognitive science field. Now, this has opened the field of linguistics up to a more rigorous scrutiny, and the reality that much of our linguistic system is invisible to us has frustrated many adjacent scientific fields. Linguistics occupies a rarer spot in sciences, as it has a unique susceptibility to people outside the field assuming they know about language, as everyone uses language. Now that linguistics participates in a wider scientific realm, it is more subject to interpretation from computer scientists, psychologists, neuroscientists, etc. One example of this will be discussed later in the paper by Steven Piantadosi on large language models (LLMs). Again, despite many informative and innovative influences between linguistics and computer science, it will become apparent in this later discussion on LLMs how the interpretation of data in different fields will always remain different.

In a time where we greatly understand computers but not nearly as much of the human brain, the inadequacy of FLT in the early years of the cognitive revolution are apparent in Chomsky's own works. In *Aspects of the Theory of Syntax*, he already addresses the problem of the phrase-structure grammars. When speaking about the adequacy of theories to be tested using empirical data, he mentions the desire to rule out as many theories as possible so that we may focus on the promising ones, mostly on the basis of "weak and strong generative capacity". To be precise, the study of strong generative capacity is related to the study of descriptive adequacy and grammars are descriptively adequate if they strongly generate the correct set of structural descriptions (Chomsky, 2014). The book then goes on to talk about viewing natural language as potentially falling within the realm of context-free, or even more powerfully context-sensitive languages: " [...]

these limitations in strong generative capacity carry over to the theory of context-sensitive phrase-structure-grammar"(Chomsky, 2014) What is being emphasized here is that natural language may not be as simply described as any of these categories. It may have a generative capacity as powerful as Turing machines, a type of infinite capacity description that makes the task of determining the rules of natural language more difficult.

However, early FLT inquiry being inadequate in some respects and not so interesting to general linguistic inquiry (at least in Chomsky's case), didn't stunt the longevity of FLT as we now know. As will be presented in the next section, Stuart Shieber's work on Swiss German is an important work in the context of FLT. Despite the many differences in computers and the human brain, FLT in the context of both of these disciplines will remain important for many years to come.

4 Use of Formal Language Theory as Framework in Later Works

Although Chomsky's explicit categorization of phrase structure grammars led to his rejection of their power, expressing that "it was a mistake, in the first place, to suppose that the base component of a transformational grammar should be strictly limited to a system of phrase structure rules" (Chomsky, 2014), future linguistic investigation and his own following works would prove that formal language theory did have some longevity in some ways: "such a system does play a fundamental role as a subpart of the base component." (Chomsky, 2014) Specifically, Stuart M. Shieber's 1985 work on Swiss German titled "Evidence Against the Context-Freeness of Natural Language" (which was also complimented by other works at the time by Riny Huybregts and Christopher Culy) (Jäger & Rogers, 2012) works well within the framework of FLT. It uses constructions in Swiss German to analyze the cross-serial subordinate clause, to make the argument that some aspects of natural languages can cross the "context-free barrier" (Shieber, 1985), into the context-sensitive territory.

In terms of raw data, the Shieber text uses as a main point the concept of cross-serial dependencies. Dependencies in and of themselves are nothing new, often the example of either/or/neither/nor is given to show this concept in English. It ties two elements that are not adjacent to each other, where the utterance of one is dependent on some further away element. For example, the equation $a^n bcde^n$ is completely arbitrary but the e variable depends on the a variable without being directly next to it, a quite simplified version of the concept. Moreover, if you nest these dependencies within each other, English seems to allow for an unlimited number of them, as long as they remain nested (Jäger & Rogers, 2012). The notion of *unlimited* already ruled out the possibility of natural languages being regular, as since we saw earlier, regular languages (represented by a finite-state automaton), cannot be unlimited. However, given this example of Swiss German data, taken from Shieber's article:

...*mer d'chind	de Hans	es Huus	lönd
... we the children-ACC	Hans-ACC	the house-ACC	let

hälfe aastriiche
help paint

'...we let the children help Hans paint the house'

A few things to note: firstly, the case marking in Swiss German between the verb and its object must match. However, unlike English, the verbs and their objects are not necessarily adjacent, so we find this type of dependency (cross-serial). This type of phenomenon that occurs in Swiss German is not nested and potentially can happen a large if not unlimited amount of times. So, as Shieber discusses, the context-free category is no longer powerful enough to describe this language, since context-free can handle unlimited dependencies like this as long as they are nested only as mentioned. (Shieber, 1985)

The implications of this type of research in the 1980's can lead us to the discussion of sub-categories within the next step in the Chomsky hierarchy, that is, if natural languages cannot be properly described by context-free languages, we must move to the more encompassing context-sensitive category. But where within this large category could natural languages lie? Firstly, it's important to note that just because certain aspects fall outside of regular or CFL's, this does not mean that all aspects of an entire language must use the full complexity of, for example, context-sensitive languages (Jäger & Rogers, 2012). What follows from this and other research was a subdivision of what is called "mildly-context-sensitive languages" into a few parts, namely (in ascending order of power): the context-free languages, Aravind Joshi's Tree Adjoining Grammar (TAG) and Chomsky's Minimalist Grammars and finally the truly context sensitive. (Jäger & Rogers, 2012) What seems to be the conclusion discussed in (Stabler, 2004) is that TAG's are sufficiently powerful to describe the kind of phenomenon that we described in Swiss German, where we declared context-free languages too weak. This puts us somewhere in between context-free and context-sensitive. Although where natural language falls on this scale seems to be of little interest to some, the answer only truly matters if we can show that natural language is not some capacity that requires infinite power. If indeed natural language is not described by the Type 0 unrestricted grammar, recognized by the Turing machine, then the restrictions can eventually be described. Finding out where natural language falls on this scale does not inherently create us a model that describes linguistic competence. However, pinpointing the correct category gives a better idea of the scale of restrictions that a model would have to work within to be a good model, one that makes predictions and is falsifiable. In the short term, this line of linguistic inquiry may not seem immediately relevant, but "big picture ideas" often prove to be useful somewhere down the line.

Shieber's research mirrors the conclusion found in the late pages of *Syntactic Structures*. As previously mentioned, Chomsky had come to a pretty certain conclusion on the inadequacy of the finite state grammars at describing natural language. However, he had not completely discounted the potential future relevance of the categories described (like regular, context-free, etc.) In the same way, Shieber finds an important syntactic phenomenon with implications of the extra-context-freeness of natural language. Like Chomsky, he does not discount the usefulness of FLT as a whole. Simply, more advancements in the field will eventually lead back to the "big picture" question of which category properly describes natural language. Although the Swiss German data is only one example, the fact that linguistic investigation has, at points, led back to the question of whether or not natural language is context-free, means there is still relevant information to discover on this subject.

In the conclusion, Shieber explains:"though the search for tight formal constraints on grammars and restrictive mathematical properties of natural languages (in the spirit of the context-free hypothesis) is a worthy goal, the present research may be a clue leading in a slightly different methodological direction. [...] The search for formalism restrictions should therefore be accompanied by research on precise models of language mechanisms" (Shieber, 1985) What is being

explained here is that because of the exceptions found in this data, we can no longer assume "tight formal constraints" without more research on language mechanisms. Once the language mechanisms are better understood given what we know from the exceptions found by Shieber, then the discussion can go back to the exact restrictions of natural language. In other words, more precise work must be done first before the greater discussion can be had.

On the subject of the appreciation for mathematical properties of natural languages, it seems the enthusiasm for them was never truly lost. Although the following quote is in the context of Minimalism, it illustrates the continued desire for formalization: "But formalization also extends the reach of syntactic theory. Mathematics is a powerful lingua franca, a tremendous bridge builder. By putting Minimalism on a mathematical foundation, one can link it to existing work on parsing and learnability (see Sec. 5). Not only does this strengthen the connection between theoretical syntax and psycholinguistics, it also opens up the gate to largescale applications in modern language technology." (Graf, 2021) What Thomas Graf expresses in the quote shows the longevity of formalization efforts like FLT in linguistics. Although the next section of this paper discusses Large Language Models (LLM), which themselves do not use Formal Language Theory, I will introduce here a bridge between these two sections.

Not obvious at first glance, considering LLM's are, in simplified words, statistical models, but FLT has made its way into the field. (OpenAI, 2019) In a small, but perhaps in the future significant way, some PhD candidates at Rutgers University have proposed a way to integrate an approach based on Formal Language Theory into LLM's to increase performance. By allowing the advantages of natural languages (viewed in the framework of FLT) to be integrated into this system, they believe they can advance the model to a higher level.(Li, Hua, Wang, Zhu, & Zhang, 2024) This isn't the proper paper to go into the technicalities of what is an admittedly complex modification to an existing AI model. However, the simple idea of using FLT in a language model shows the power of formalization in the way Shieber, Thomas Graf, and many before and after them have mentioned. Since the Formal Language Theory we discuss in linguistics exists in a generative framework, I find it both relevant and interesting to combine, so to speak, many of the aspects presented so far. Formal Language Theory, computer science, linguistics, the generative framework, all come together when discussing some of the modern papers that have been plentiful since LLM's have come onto the scene.

5 Some Modern Views of Generative Grammar

Introduction on The Advent of Large Language Models (LLM)

An extremely popular paper by Steven T. Piantadosi has been a hot topic of conversation, published in March 2023, when ChatGPT was on the minds of everyone in the know. The paper, which should have remained solely a discussion of the implications of LLM's on linguistics, goes beyond this. I say *it should have* since the extrapolations it makes somewhat delegitimize what is otherwise an interesting paper on LLMs, and those extrapolations will be addressed in the following subsections. It suggests that large language models serve as an amazing model of a theory of language (Piantadosi, 2023), and by extension human cognition (as language is often viewed as such). The paper seems to also take significant time personally addressing issues the author has with generative grammar theories, and even personal, unwarranted comments on Noam Chomsky himself, claiming his: "[...] remarkable downfall" and the "cautionary tale about what happens when an academic field isolates itself from what should be complementary endeavours"(Piantadosi, 2023).

The remark that the field of linguistics has isolated itself is just another example of the unique susceptibility I've previously described of linguistics. This unique susceptibility can be summed up as the desire for people to have opinions on language, because the fact that everyone uses it feels like qualification enough to pronounce on it. The paper presents massive misunderstandings about the nature of language and cognition, which were addressed by Roni Katzir's paper and Jon Rawski and Lucie Baumont's short response. Aside from these responses, which serve as directed refutations, it seems both engineers and linguists agree that we should proceed with caution when estimating the relevance of a LLM on a theory of language or cognition. A paper by engineers at Apple,(Mirzadeh et al., 2024), and a linguistic one by Heuser et al. (Heuser, Yang, & Kodner, 2024), are more careful with their analysis of the potential that can be extracted from the LLM. The concepts and arguments presented by Piantadosi, their refutations, as well as the engineering and linguistic papers mentioned above will be summarized in this section. This further adds to the thesis arguing for the relevancy of generative grammar even in the face of modern advances in fields which intersect with linguistics e.g. neuroscience, psychology.

Steven Piantadosi's Paper: "Modern language models refute Chomsky's approach to language"

First, it is important to be able to summarize some key points made in the original paper before discussing in more detail why they may be misguided. First, a neural network is part of the field of machine learning. The neural network is inspired by the functioning of the brain, more specifically the concept of action potential. Once the neuron of the brain reaches a voltage difference that crosses a threshold, it fires off an electrical signal. In the same way, the structure of the neural network has a threshold for its "nodes", and once it is crossed it sends information to different layers (IBM, 2023). The LLM, or large language model, is an application of this concept (OpenAI, 2019). It is a neural network. Overall, the paper tries to make the argument that LLM's serve as proof that generative grammar's approach to language has been wrong all this time. According to the paper, the "models form probabilistic expectations about the next word in a text and they use the true next word as an error signal to update their latent parameters" (Piantadosi, 2023). That is a simple explanation of the function of the GPT which is the subject of this paper. There are also claims that the computer model can "integrate semantics and syntax" (Piantadosi, 2023). Overall, this paper makes for a great way of selling the LLM as a technical marvel. It can memorize and imitate really complex things about language that we did not think previously possible. In this enthusiasm seems to have been lost some facts about human language acquisition. The text tries to get ahead of possible criticisms (which in fairness it does admit to in some part) by first highlighting in what way the author thinks generative theories have completely failed. It is not an extensive "exposé", nor is it expected to be, but it fails to compel in any meaningful way why "approaches from generative syntax are not competitive in any domain and arguably have avoided empirical tests of their core assumptions"(Piantadosi, 2023). There is of course mention of the sheer size of datasets the models are trained on, and an admittance that this clashes with what we know regarding the *Argument for the Poverty of the Stimulus*, but this does not seem to affect in any way the argument being made.

There is a large emphasis on humans being able to infer from the data large parts of what allows grammar. This argument has existed for a few decades now, and the argument against it is summarized in the preface to the most recent edition to *Aspects*. Regarding the excitement over infants using statistical learning for symbol sequences: "even if Saffran, Aslin, and Newport's

results had provided some significant support for much earlier proposals about word boundaries within generative grammar, they would have told us virtually nothing about whether humans 'use generalized statistical procedures to acquire language.'" (Chomsky, 2014) This is a good example of a piece of data, research which concludes something about the way symbol sequences are learned, but was without further questioning applied to language. It was at the time, and still today, too soon to apply the conclusion of this study to the linguistic competence of humans. In summarizing Chomsky's argument, data regarding the fact that children utilize statistical learning for symbol sequences does not mean in any way that this is entirely how we learn language, and instead should be seen as a probable learning method that can still factor in overall into the generative grammar theory, and is in no way a rebuttal of the *Argument for the Poverty of the Stimulus*.

A Summary of Roni Katzir's Response to Piantadosi

In his article entitled "Why large language models are poor theories of human linguistic cognition. A reply to Piantadosi (2023)", Roni Katzir responds to the previously discussed paper by testing the GPT on a number of subjects that could be achieved by almost any human. Although much of this article will be summarized, the phrase "discovering that they [LLM's] teach us about how humans work would be startling indeed, akin to discovering that a newly designed drone accidentally solves an open problem in avian flight" (Katzir, 2023) perfectly encapsulates the fundamental problem with the ideas Piantadosi presented.

The first point made is in response to the claim that these models could inform something about human cognition, then the models must display linguistic behavior similar to that of humans. The GPT in this response is tested in numerous ways to display its inadequacy. The generalization about conjuncts, that leaving a gap in one necessitates leaving a gap in the other (like the example "The person that Mary met yesterday and that John will talk to Ed about tomorrow arrived.") (Katzir, 2023) The program thought that because it overall had less information it must be inferior, despite the fact that the sentence containing more information violated a fundamental rule. Children, as Katzir explains, instantly know this rule to be true, meaning they are prone to knowing which sentence is more correct. In this case, the computer has decided to opt for extra information in place of correct syntax, and this was proven by testing four **different** LLM's. The same argument was made by testing, according to Katzir's paper, the reversal of order between between the subject and the auxiliary verb in a study by (Yedetore, Linzen, Frank, & McCoy, 2023). In short, it is not that LLM's cannot be made to resemble human linguistic capacity, but rather that their biases are so different than ours that claiming they can be a serious theory of cognition is wrong.

Skipping over a few more examples, the competence vs. performance distinction is addressed. Competence being distinct from performance is not a controversial statement, but rather a concept which should have factored in to the conclusions made in Piantadosi's paper, but did not. For example, Katzir's paper gives the example of center embedding being of confusion to lots of people, but given a little extra time to process, usually no mistakes are made in comprehension. In a kind of human version of computer *cache*, the fact that a person needs some extra time is not a question of not knowing center embedding, but just a memory limitation. The same cannot be said of the LLM; "their behavior directly reflects their competence" (Katzir, 2023). The competence performance distinction is a useful tool for showing that although humans and the programs can make similar mistakes and sound similar, the deeper reasons for these are clearly not the same. Errors being a matter of competence vs being a matter of performance is a huge insight as it always has been, since it tells us something about how our brain's instincts, intuitions, innate knowledge works.

Training on datasets vs training based on innate structures is completely different. Computer programs can absolutely be made to imitate humans, but Katzir's paper goes over the important point that LLM's consistently choose correctness based on likelihood. In an example employing the sentence "The little duck that met the horses with the blue spots yesterday...", the program is then asked to choose between the next word being "are" or "destroys". The computer chooses are, even though the subject the little duck and the present verb tense in "destroys" are correct, simply because it decided that "are" is more likely and doesn't change the overall feeling of the sentence. It cannot help making generalizations of this sort.

There are a few other small examples, but the important point made by Katzir can be summarized in the footnote he leaves regarding Piantadosi's suggestion that probabilities are good in linguistic models: "But he confuses two possible roles for probabilities: as part of linguistic knowledge and as part of the learning model." This ties back to the earlier mention of *Aspects*, in which Chomsky discusses the possibility for statistical type learning methods to still fit within the generative theory. Katzir is saying that just because statistics can provably show certain parts of our learning development, this does not overall affect the fact that linguistic knowledge must not be a matter of probabilities, but a matter of innate structures, as has been shown over the years.

A (Brief) Summary of Jon Rawski and Lucie Baumont's Response to Piantadosi

Rawski and Baumont's response to Piantadosi takes a different approach, opting to avoid using practical examples (as this had been done) to refute the claims made in the paper. Instead, it chooses the logical inference route. The paper uses MLM as the acronym for modern language models, but it refers to the same model as LLMs. To summarize, the text uses Guest and Martin's argument about the validity of cognitive modeling. The claim P says "MLMs are a theory/model of human linguistic capacities, or 'do what humans do'", and the claim Q states "MLMs correlate with/predict human behavioral and/or neuroimaging data" (Rawski & Baumont, 2023). If P were to imply Q, which is something which needs proving itself, this would be a fine hypothesis to start off on. However, Piantadosi instead decides that since P *implies* Q, Q *therefore* P is equally true. This is wrong, and is a famous error called "the fallacy of the converse" in logic. Rawski's paper then goes on to explain that only type of logical argument we can truly make is that if the computer models fail to properly imitate humans, which here we would write as not Q ($\neg$Q), then we can infer $\neg P$: $\neg Q$ *therefore* $\neg P$. MLMs are thus not good theories of human linguistic capabilities (Rawski & Baumont, 2023). And in Piantadosi's case, we do see this. As explained earlier in Katzir's paper, and now implied in Rawski's response, MLMs are not only a bad account of human linguistics and cognition as a whole, but even fail in replicating basic language tasks that can be done by any human. If the main claim were to be summed up in one sentence "Explanatory power, not predictive adequacy, forms the core of physics and ultimately all modern science" (Rawski & Baumont, 2023).

How to Gain Insight from LLM's

Engineers at Apple and the linguist contributors to the following papers represent a different type of retort that is just as relevant as the ones we've seen so far. They are not direct replies, and do not criticize as harshly the possibility of gaining some use out of LLM. However, what they share in common is more obvious: a skepticism about the implications of LLM on one hand for

intelligence/cognition as an AI, and in the other article, for intelligence/cognition in humans.

In a paper entitled: "GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models", the authors describe a benchmarking issue with the way LLM's are currently described. Although this article has little linguistic relevance, the authors make clear that they do not believe that Large Language Models are capable of any logical reasoning, which is the point I want to focus on. (Mirzadeh et al., 2024) Lack of the capacity for logical reasoning certainly does not seem like a good starting point if one wants to extract a theory of the unique human faculty for language, whether you believe in generative grammar, usage-based grammar, etc. The evaluation for the mathematical reasoning capabilities of the model is usually done by a dataset used as a benchmark called GSM8K, which stands for "Grade School Math 8k". (Mirzadeh et al., 2024) The paper's authors propose a new benchmark, as they believe the GSM8K may be too easy, giving the impression of more impressive performance. This engineering paper does not address linguistics, but concludes that the model cannot reason. The model's performance falls dramatically when confronted with any amount of superfluous information, and even in the case of examples given by the user to help the model, it still fails quite badly. (Mirzadeh et al., 2024) Of course, being engineers, the paper is still optimistic about the possible uses of LLM's, but for our interests, the failure of the model to even imitate reasoning is quite telling. Extracting from their conclusions, it is safe to say that attributing to a still underdeveloped software the possibility of a theory of language may be "over-hyping" it.

Now, on the linguistics front, the paper: "Evaluating the Existence Proof: LLMs as Cognitive Models of Language Acquisition" by Heuser et al. has a similar structure and approach as the previous one. It begins with a summary of LLMs, and how the authors believe the environment for training them is different than children. Next, it talks about benchmarks, like our previous paper. Approached from a different angle, these authors reach similar conclusions as the engineers at Apple: the unearned success of these benchmarking tests may be contributing to the excitement surrounding all the possibilities for the AI. (Heuser et al., 2024) They propose to use a linguistic-oriented dataset called LI-Adger, by Sprouse et al., 2018 . In their own words: "LI-Adger sentences were hand-constructed by theoretical linguists who are well aware of the contribution of non-grammatical factors such as lexical frequency, sentence parsing strategies, and semantic and pragmatic plausibility to the behavioral measure of acceptability rating" (Heuser et al., 2024) This dataset attempts to alleviate some of the factors that may not have been considered by the programmers of the AI, to mirror performance better to human performance. Regardless of all these steps taken to try to imitate the conditions of a person learning language at an early age, LLMs often still cannot perform in a way that is considered close to humans. Much has been said by linguists about the size itself of the training datasets (Heuser et al., 2024). This is a good observation, of course, to point out that these models are trained on much larger amounts of words and sentences than a child would. However, even this problem, addressed in the BabyLM challenge for example, makes clear the following: "not all aspects of linguistic knowledge are learned at the same time: the developmental trajectory of language acquisition matters as much as the final grammar attained." (Heuser et al., 2024) So even if we were to one day be able to develop a Large Language Model which can, on a similar amount of data than a child, give us a semblance of human language, this does not elevate the AI to suddenly being an insightful way of investigating the human language faculty.

To conclude this section, the famous quote attributed to Frederick Jelinek comes to mind: "Every time I fire a linguist, the performance of the speech recognizer goes up" (Exact source

unknown). It may be true that linguists are a nuisance on performance of computer models, but I think many who want to extract meaning about ourselves from the Large Language Model are not seeing the forest for the trees, so to speak (I would consider a theory of language to be the forest, here). There may be more to gain by continuing to treat AI as a tool, and focusing on humans for theories of human language.

6 Conclusion

To summarize, Formal Language Theory remains relevant in linguistics, computer science, philosophy of the mind, and many other disciplines with its appealing abstraction of innate structures. The desire to explain one of the mysteries of humanity, the linguistic capacity, drove the development and subsequent refining of FLT, and in a greater context, parts of generative grammar itself.

In the case of computer science, it is no surprise that a mathematical-based model was a perfect fit for abstracting a computer's functioning at the basic level. If one were to make the most uncontroversial statement of this paper, FLT has most definitely earned its spot as a permanent fixture of theoretical computer science as we've known it. To draw a comparison, in the same way that quantum computing may one day come to replace today's devices, linguistic theory's fate will no doubt evolve. Going through a historical overview of FLT in linguistics is useful not only for laying out the answers it provides, but for describing the new questions it proposed. Where does natural language fall? Context-freeness or mild-context-sensitivity? These questions are still relevant.

Throwing a wrench into the works of generative grammar has been the goal of many modern papers regarding linguistics, due in part to the power of its implications. This is healthy scientific discourse, but it's easy to see how skepticism can quickly turn into obvious gaps in logical reasoning like in the case of Steven Piantadosi's article on LLMs. The argument for the validity and relevance of generative grammar stems from its empirical support, its ability to predict outcomes in various subfields of linguistics, and its natural appeal to philosophy as a whole. The clear over-enthusiasm of new scientific advancements which are prematurely applied to linguistics have long been used as proof that generative grammar is out of style. This often leads to the creation of a "strawman" version of generative grammar which is easier to attack, and serious errors are made this way. Statistical learning methods and the like will always have compelling empirical support, but do not truly strike at the innate structures proposed, as they can both coexist in a same theory of human linguistic capability.

In this paper, I summarized some introductory material on Formal Language Theory both in Computer Science and Linguistics. Contrasts between the two on a surface and on a deeper, more theoretical level were then addressed. Then, Shieber's Swiss German research and some FLT-oriented advancements in AI were used as examples arguing for the application of FLT in linguistics, and large language models. Finally, a summary of Steven Piantadosi's paper "Modern language models refute Chomsky's approach to language" and many criticisms of this paper were discussed. I personally argued against the conclusions made in Piantadosi's work. All these points come together to make an argument on the relevancy of FLT. Also, this paper seeks to clarify the importance of how proper conclusions should be drawn from data, by summarizing both arguments against the Piantadosi paper, and the work done to properly evaluate the competence of AI in the Mirzadeh et al. and Heuser et al. papers. These serve as good examples of theoretical misunderstandings leading to faulty conclusions.

References

Chomsky, N. (2002). *Syntactic structures*. Mouton de Gruyter.

Chomsky, N. (2014). *Aspects of the theory of syntax* (No. 11). MIT press.

GeeksForGeeks. (2023). *Chomsky hierarchy in theory of computation.* Retrieved from `https://www.geeksforgeeks.org/chomsky-hierarchy-in-theory-of-computation/` (Accessed on July 19, 2023)

Graf, T. (2021). Minimalism and computational linguistics. *Lingbuzz/005855*.

Heuser, H. J. V. M. A., Yang, C., & Kodner, J. (2024). Evaluating the existence proof: Llms as cognitive models of language acquisition.

IBM. (2023). *What are neural networks?* Retrieved from `https://www.ibm.com/topics/neural-networks` (Accessed on September 30, 2023)

Isac, D., & Reiss, C. (2013). *I-language: An introduction to linguistics as cognitive science*. Oxford University Press, USA.

Jäger, G., & Rogers, J. (2012). Formal language theory: refining the chomsky hierarchy. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *367*(1598), 1956–1970.

Katzir, R. (2023). Why large language models are poor theories of human linguistic cognition. a reply to piantadosi (2023). *Manuscript. Tel Aviv University. url: https://lingbuzz. net/lingbuzz/007190*.

Kemmerer, D. (2015). *Cognitive neuroscience of language*. Psychology Press.

Li, Z., Hua, W., Wang, H., Zhu, H., & Zhang, Y. (2024). Formal-llm: Integrating formal language and natural language for controllable llm-based agents. *arXiv preprint arXiv:2402.00798*.

Linz, P., & Rodger, S. H. (2022). *An introduction to formal languages and automata*. Jones & Bartlett Learning.

Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., & Farajtabar, M. (2024). Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

NASA. (2015). *Imagine the universe!* Retrieved from `https://imagine.gsfc.nasa.gov/science/objects/milkyway1.html#:~:text=It%20is%20very%20difficult%20to,is%20about%20100%2C000%20light%20years.` (Accessed on August 25, 2023)

OpenAI. (2019). *Research - better language models and their implications.* Retrieved from `https://openai.com/research/better-language-models` (Accessed on September 30, 2023)

Piantadosi, S. (2023). Modern language models refute chomsky's approach to language. *Lingbuzz Preprint, lingbuzz*, *7180*.

Rawski, J., & Baumont, J. (2023). Modern language models refute nothing. *Lingbuzz Preprint*.

Shieber, S. M. (1985). Evidence against the context-freeness of natural language. In *The formal complexity of natural language* (pp. 320–334). Springer.

Stabler, E. P. (2004). Varieties of crossing dependencies: structure dependence and mild context sensitivity. *Cognitive Science*, *28*(5), 699–720.

Yedetore, A., Linzen, T., Frank, R., & McCoy, R. T. (2023). How poor is the stimulus? evaluating hierarchical generalization in neural networks trained on child-directed speech. *arXiv preprint arXiv:2301.11462*.