CS 4510 Automata and Complexity	January 30th 2023

Lecture 6: Syntactic Structures

Lecturer: Abrahim Ladha *Scribe(s): Akshay Kulkarni*

1 Syntactic Structures

1.1 Language

Linguistics is the study of language. This includes its structure, etymology, history, everything. Its systematic. Fundamentally, language is communication. Suppose we have a listener and a speaker, and the speaker says "horse". The way we represent that idea syntactically, as a word or utterance, can have nothing to do with the idea itself. That's language. Language is an agreement we all have about what things mean. "Horse" is nothing. "Horse" is an utterance I make by moving vocal cords and exhaling air at a frequency you pick up with your ears.

The concept, the semantic value of "horse" is vastly more complicated than the sound or word itself. I say or utter "horse" and in our mind, you create an image of a herbivore quadruped, maybe its brown. From an information theory perspective, this is a lossy channel, as the idea contains more information than the word.

While animals have certainly demonstrated ability to understand and mimic language to a reasonable extent, humans are the only beings capable of implementing complex language. You can talk to animals, but they won't talk back. Some argue that language is central to humanity's evolutionary identity. As a species, or ability to communicate ideas is exactly what makes us human. It wasn't the upright posture, larger prefrontal cortex, opposable thumbs, toolmaking, no. It was the ability for us to come to consensus and work together on complex tasks. If I, as some paleolithic ape generate the idea "I go hunt mammoth". This idea is totally worthless, I go hunt mammoth the outcome is I get trampled. Instead I communicate this idea and now its "we go hunt mammoth", suddenly its more serious. First we took down a mammoth, and second, we built a computer. Theres some steps in between those two.

Since all ideas must be expressed as language, it was an old-world view that the study of language itself was the only way to study ideas. The study of language was the study of everything. One of Chomskys accomplishments was to help separate these two. Syntax, the structure of language, and semantics, the meaning of language, are not interchangable.

1.2 Syntactic Structures

The importance of Chomsky's short monograph was not that he solved language in general, but rather he came into someone elses house with more math than them. He came into a very empirical field, and brought in an as theoretical as possible perspective. Using relatively simple intuitive arguments, he was able to make true generalizations about what is an incredibly complex system.

1.3 Chomskyan view of language

Consider a baby. It it not born speaking any language. Googoo gaga and so on. Totally unintelligible. Although a baby is not born knowing any language, it somehow knows how to learn a language. Airdrop that baby into a group of people speaking a language and as it mentally develops, it will learn how to speak among them. This would be independent of anything about the structure of the language. It would not necessarily learn in school what nouns and verbs are in order to speak. Syntax is an innate aspect of language determined by a "Universal Grammar". There are biological conditions which shape the structure of language. The way our brains have the wires and pipes cause limitations in what possible structures language must take.

To study language, it is okay for us to limit ourselves to english. Languages share many universal features. For example, delimination with a space. Have you ever thought about why sentences come in lists and not trees or some other structure in which may not have a kind of topological sort? All grammatical operations appear to be binary as well. The set of grammatical sentences appears to be infinite L, but appears to be contructed recursively from some sort of finite atomic pieces, like a basis. In english, that would be our alphabet, Σ. Secondly, for any english specific artifacts we may encounter, there are almost certainly analoguous issues in other languages.

As we develop a theoretical model for grammar, it is sufficient for us just to short the ability to distinguish the grammatical from the ungrammatical. Such a device or structure which can help us separate these two, can also help us generate new grammatical sentences.

1.4 The Independence of Grammar

Consider the following two sentences:

1. Colorless green ideas sleep furiously.

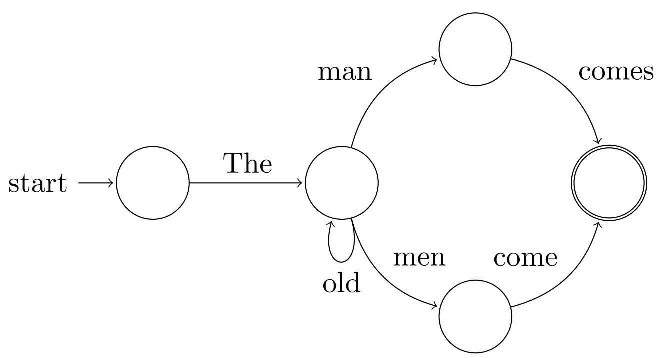
2. Furiously sleep ideas green colorless.

Let us study the first sentence. Certainly it is grammatical. Somehow in our brain exists a distinguisher, and we can read this sentence and come to the consensus that it is grammatically correct. Next of note is that the sentence is totally devoid of meaning. What would the subject the sentence be? Ideas? and they somehow can sleep? and do so furiously? Its colorless, yet green? It has no semantical value, and does not communicate any idea (besides perhaps confusion). This is a kind of counter example, and forces us to separate syntax, the structure of language, from semantics, the meaning of language.

Second, note that the first sentence is grammatical, but the second one is not. The first, simply by intonation and word pattern seems comfortable. It would be easier to remember and recite. The second is ungrammatical, and troubling. Yet, these two sentences, the frequency of sequential word choices is equivalent because one is a word reversal of the other. These two sentences have been uttered equally likely in all of english, that is, a negligible amount[1]. This is our second observation. The ability of a syntactic structure to distinguish the grammatical from the ungrammatical must be independent of the sentences proximity to english. The first being grammatical, and the second not. Any model based on probability may be unable to distinguish these two based on this kind of frequency alone, but we argue, must be able to. It is also quite likely that sentences you come across have never been said before. Even the sentences you are reading now. Something like 15% of Google's daily searches have never been searched before.

1.5 English Contains some Regular Substructure

Natural languages are not formal formal languages, but we can apply similar arguments. Here we show a substructure of english has some similar structure to a regular language. For example, the following DFA[2] can be used to model the formation of a substructure of english:



We are not concerned with the study of finite languages, but of infinite ones decidable by finite structures. Here, this decides an infinite language because it has a point of recursion. It may be inappropriate to describe someone as "old old old old...", but it is not ungrammatical, it is a hyperbole.

1.6 English is Not Regular

First, recall our three non-regular languages

- $\{a^n b^n \mid n \in \mathbb{N}\}$
- $\{ww^R \mid w \in \Sigma^*\}$

[1]It is ironic that Chomsky chose this sentence because it would be statistically infrequent, but by choosing it as an example, he has made it very famous. See its own wikipedia page

[2]Chomsky calls this a Finite-State Markov Process (FSMP)

- $\{ww \mid w \in \Sigma^*\}$

We show by an analogy, that there does not exist a DFA for a substructure of english.

Proof. Let $S_1, S_2, ...$ be declarative sentences. Let S be the sentence "if S_i, then S_j". There is no reason we may not substitute S into S_i Observe that upon repetition of this n times, we get

$$\text{“(IF)(IF)(IF)(IF)} \ldots S_i \text{ (THEN)(THEN)(THEN)(THEN)} \ldots S_j\text{”}$$

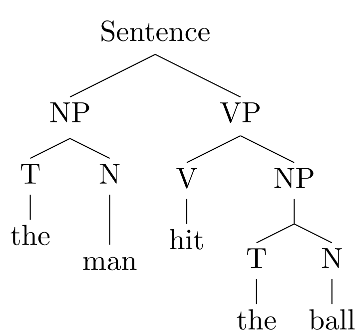
can we written as $(\text{IF})^n S_i (\text{THEN})^n S_j$, this is quite similar to our first known canonical non-regular language. We proved that language was non-regular by pumping, and similarly here, there would not exist a DFA for this substructure of english. Another good example is the Dyck language, the set of balanced parenthesis. One would recognize if a sentence had unbalanced paranthesis and distinguish it as ungrammatical. An an example, consider "((hello there.)(((". □

1.7 Phrase Structure

It had been known for centuries of the recursive nature of language. How sentences can be built from fragments, fragments from words, words from letters, and so on. Sentences have a hierarchical structure, and this structure is governed by the rules from grammar. Chomsky formalized these observations to justify what his next model of study was, and why it was ideal. He defines something called a phrase structure, which is a generalization of what we now call a CFG. For now, lets suppose phrase structures are just CFGs. We can remark that this device is incredibly useful as generative model for language. Consider the following model:

$$\begin{aligned} \text{SENTENCE} &\to \text{NOUN PHRASE} + \text{VERP PHRASE} \\ \text{NOUN PHRASE} &\to \text{ARTICLE} + \text{NOUN} \\ \text{VERB PHRASE} &\to \text{VERB} + \text{NOUN PHRASE} \\ \text{ARTICLES} &\to \{\text{a, the, etc.}\} \\ \text{NOUN} &\to \{\text{man, men, ball, etc.}\} \\ \text{VERB} &\to \{\text{hit, took, etc.}\} \end{aligned}$$

Here, we have a phrase structure for a declarative substructure of english. A production of a sentence from our phrase structure can be expressed as a parse tree.



Its notable here that a parse tree gives less information than a list of productions, as just from the tree, you don't know what order the rules were applied in.

1.8 Limitations of our Phrase Structure

Although we note that this generates grammatical sentences, it can also generate ungrammatical ones. This example is with respect to singular and plural words.

1. "The man hit the ball."
2. "A men hit the ball."

The second sentence is clearly not grammatically correct.

> Chomsky: "We must be able to limit the application of a rule to a certain context"

A context-free grammar is quite literally, free of context. If you have a production of $N \to \{$ nouns$\}$, then you can substitute in any noun. Like mad libs, it may not be grammatical. We want to consider applications of rules which are sensitive to context. A production can only be applied if conditions are met on the part of the working string before and after the substring we would insert. Comparison of language models:

Model	Example rule
Regular grammars	$A \to bE$
Context-free grammars	$A \to bCdEf\ldots$
Context-sentitive grammars	$xAz \to xyz$

Here, $x, z \in (V \cup \Sigma)^*$. You can only make the substitution $A \to y$ when in the current working string, A is preceded by x and followed by z. These types of rules are called context-sensitive, because they are quite literally, sensitive to context. They are strictly stronger than context-free grammars, and we will not spend any more time on them. For our small piece of english we are studying, we can modify the phrase structure with context sensitive rules to solve our issue with singular and plural words as follows.

$$NP_1 \to T_p + N_p$$
$$NP_2 \to T_s + N_s$$
$$T_s \to \text{a}$$
$$T_p \to \text{the}$$
$$NP_s \to \text{man}$$
$$NP_p \to \text{men}$$

Here, N_p, a non-terminal for plural nouns, cannot be preceded by T_s, singular articles. This makes the ungrammatical production of "a men" impossible.

I highly suggest you read Syntactic Structures in full. This is a high level overview of some of the simpler and early theorems made, and how they were argued.

1.9 Further Reading

- Syntactic Structures by Noam Chomsky
- Poverty of the Stimulus
- How To Know What Words Mean - Troublehacking with Drew Cleary

2 Chomsky Normal Form

Given a word w and a grammar G, is $w \in L(G)$? This is surprisingly non-trivial. We say a CFG is in Chomsky Normal Form (CNF) if it has productions only of the form:

$$A \to BC$$
$$A \to a$$

where the capital letters are any non-terminals, and the lower-case letters are any terminals. Additionally B, C cannot be the start state. and $S \to \varepsilon \iff \varepsilon \in L(G)$. Note that obviously $\mathscr{L}(CNF) \subseteq \mathscr{L}(CFG)$. We have a process to convert any CFG into CNF form, meaning that $\mathscr{L}(CFG) = \mathscr{L}(CNF)$.

1. Add a new start State $S_0 \to S$. Now every rule will not have the start state anywhere on the RHS.
2. Delete and patch all $A \to \varepsilon$ rules. For example if you have rules $R \to uAv, A \to \varepsilon$, you now have rules $R \to uAv \mid uv$.
3. Remove all unit rules $A \to B$ (i.e. $(A \to B, B \to C) = A \to C$
4. Convert rules of length greater than two into a chain of rules as follows. $(A \to u_1 \dots u_k) \to (A_1 \to u_1 A_1, A_2 \to u_2 A_2, \dots, A_{k-1} \to u_{k-1} u_k)$
5. $\forall a \in \Sigma$, replace a with A using $A \to a$.

Steps three and four may need to be repeated many times because applying one patch may introduce a need for another.

2.1 Advantages of CNF

Lets prove that if a word of length n is produced by a grammar in CNF, it takes exactly $2n - 1$ productions. Lets work backwards.

$$w_1 ... w_n \overset{*}{\Longleftarrow}_1 W_1 ... W_n \overset{*}{\Longleftarrow}_2 S$$

- The last productions (1) goes from n terminals to n non-terminals. At each production, exactly one non-terminal is replaced by exactly one terminal, so this takes n productions.

- For (2), to go from n non-terminals to one terminal, our start terminal, requires $n-1$ productions. Every rule of a grammar in CNF takes one non-terminal, and adds two. So for each production, if non-terminals are added, a production adds exacly one.

Combined, we see that a grammar in CNF form will take exactly $2n-1$ productions to produce a word of length n. This solves our acceptance problem. Convert your grammar to CNF, compute all words produced after $2n-1$ productions, and your candidate word is in this list $\iff w \in L(G)$. Consider the following conversion of a general CFG to one in CNF.

$$\begin{aligned}
&S \to aSb \mid \varepsilon \\
&S_0 \to S,\ S \to aSb \mid \varepsilon \\
&S_0 \to S \mid \varepsilon,\ S \to aSb \mid ab \\
&S_0 \to aSb \mid ab \mid \varepsilon,\ S \to aSb \mid ab \\
&S_0 \to aX \mid ab \mid \varepsilon,\ S \to aX \mid ab,\ X \to Sb \\
&S_0 \to AX \mid AB \mid \varepsilon,\ S \to AX \mid AB,\ X \to SB,\ A \to a,\ B \to b
\end{aligned}$$

Lets verify that $\{a^n b^n \mid n \in \mathbb{N}\}$ takes $2n-1$ productions. That *aabb* takes seven productions.

$$S \implies AX \implies ASB \implies AABB \implies aABB \implies aaBB \implies aabB \implies aabb$$