# Tutorial 11: Introduction to syntax

## Syntax discussion questions

1. Speakers of any given language can produce and understand sentences they have never heard before. What does this tell us about human language?

I think this implies the structure of how language construct is not infinite and people can understand the basically meaning of language by recognizing the structure in the sentences.

1. Is the set of possible English sentences infinite? If it is not, explain why not. If it is, what does this tell us about our mechanism for generating sentences?

Everyone can speak an English sentences that didn’t exist before. So the set of possible English sentences is infinite.

It suggests that we can not use an accurate(discrete) method to analyze all the possible sentences.

1. A grammar breaks up a well-formed sentence into phrases: hierarchically structured sequences of words that ‘hang together’. When we’re building a grammar for a given language, how do we decide which word sequences should be phrases?

When we build a grammar for a given language, we need to see the statistic of the word in language. If they appear much frequently than others, then we should consider they are phrases.

1. Explain how the process of parsing a sequence of words, using a grammar, is a variety of state space search.

The initial state is blank; the goal is the sentence. When do the expansion recursively, we are constructing a tree, its leaf is the current state.

When we expand the branch, it is just like we are expanding the fringe. However, when we expand the node in the fringe, instead of checking if the node is our goal; we are checking, after we expand the node, if the tree’s leaves are matching our goal state.

3. In Lecture 21, you saw an example of a top-down parser, that searches the space of possible sentences starting with the top-level node S. Run the function top down parser demo (string, grammar) to explore how a top-down parser works.

(b)  Next, try it on a sentence that’s not parseable by the grammar. What goes wrong? How could you change the grammar so the top-down parser can correctly reject ungrammatical sentences?

It can keep going deep with infinite loop (keep generating the left branch without stop).

The reason is that the parser algorithm is depth first search, so when do the recursion it can happen that the depth is infinite. And depth first search is not complete when the tree’s depth is infinite.

Basically change the grammar rules so there is no such rule: x -> x, m, n.

## 4. Quiz: Add some new rules to the grammar, that allow it to parse/generate sentences like:

Fred thinks Jip bit the cat.

The cat thinks Fred thinks Jip ran.

A dog thinks the cat thinks Fred thinks Jip ran.





