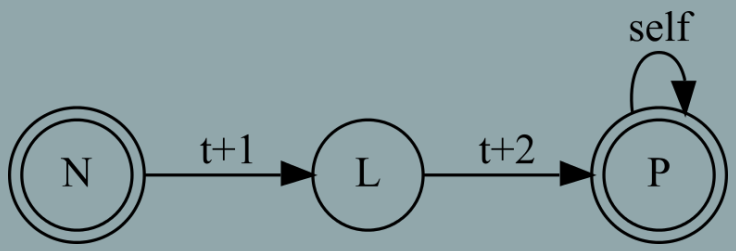
LING-362

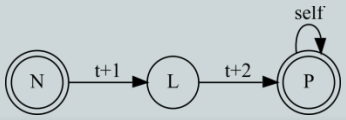
Introduction to **N**atural **L**anguage **P**rocessing

Syntactic parsing



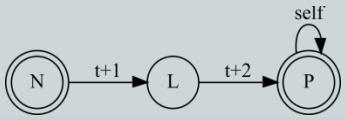
Languages and complexity

- Regular languages are the simplest grammars we can build:
  - Include all **finite** languages (where we can enumerate all expressions)
  - Potential for infinite generation (a+)
  - Optional or empty elements (ab?, ab*)
  - (Regular languages without the latter are also called **'star-free'**)



Beyond regular languages

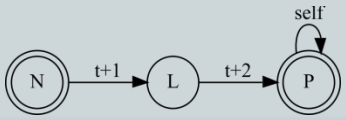
- What if we want to name a+b something else?
  - We could do things like: (DT+JJ+N)=NP: NP+…
  - This is **still** a regular language (can use FSA)
  - Even some recursion is OK:
    - x -> x
    - un + adj -> adj
- Are there constructions that can't be expressed using regular grammars?



Example: center-embedding

- In English we can center-embed relative clauses:
  - *The boy laughed*
  - *The boy the cat bit laughed*

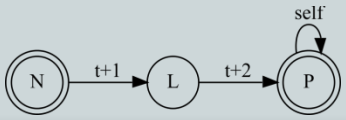
- Structure:
  - S > NP VP
  - S > NP S VP → NP NP VP VP
- We can potentially continue to center-embed…
  - Result:
    utterances of the type NP^n VP^n (or generally $a^n b^n$)



Another example

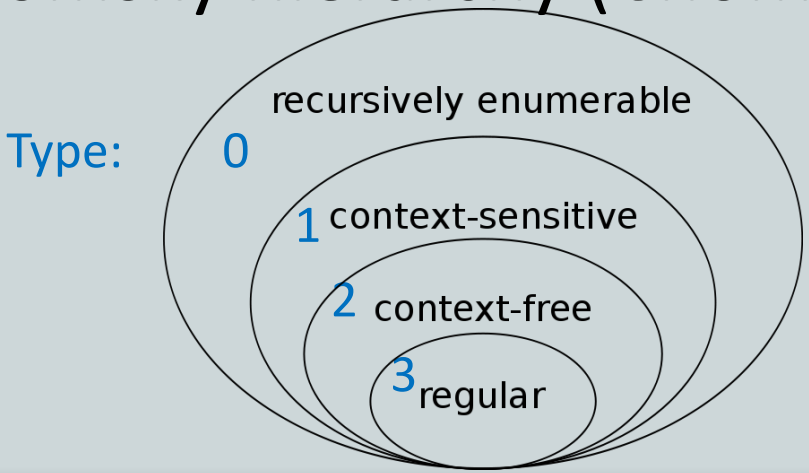
- Less famous – Semitic embedded compound modifiers:
  - [bat [melex ʻašir] yafa]
    daughter king rich.M beautiful.F
    Beautiful daughter of a rich king
  - [bat [melex [ʻam gadol] ʻašir] yafa]
    daughter king people great.M rich.M beautiful.F
    Beautiful daughter of a rich king of a great people

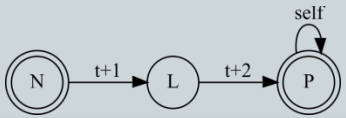
- Note that agreement information must match
- Memory: N^n A^n with matching gender/number



The Chomsky Hierarchy

- "self-embedding" categories:
  - Are a feature of **context free** languages
  - Allow us a sort of 'memory'
  - Long thought to cover human grammars
- Context free grammars (CFGs) occupy Type-2 of the Chomsky hierarchy (Chomsky 1956)

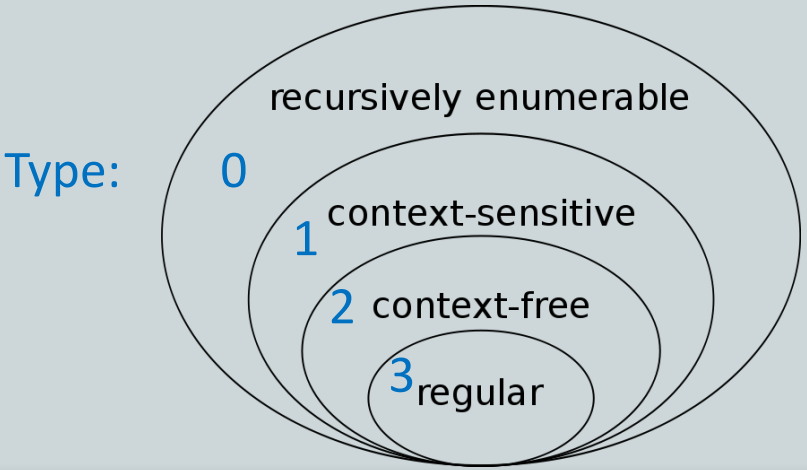


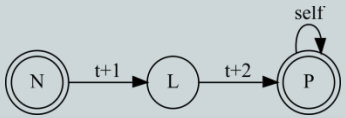


(Image: Wikimedia)

The Chomsky Hierarchy

- What level of complexity does human language have?
  - Regular grammar fits most **morphologies**
  - Context Free Grammars are enough for most **syntax**
  - Some constructions need more!
- Context free grammars (**CFGs**) occupy Type-2:





(Image: Wikimedia)

Context sensitive example:

- There are few examples of context sensitive structures in natural language
- Famous example: Swiss German crossing dependencies (Shieber 1985)

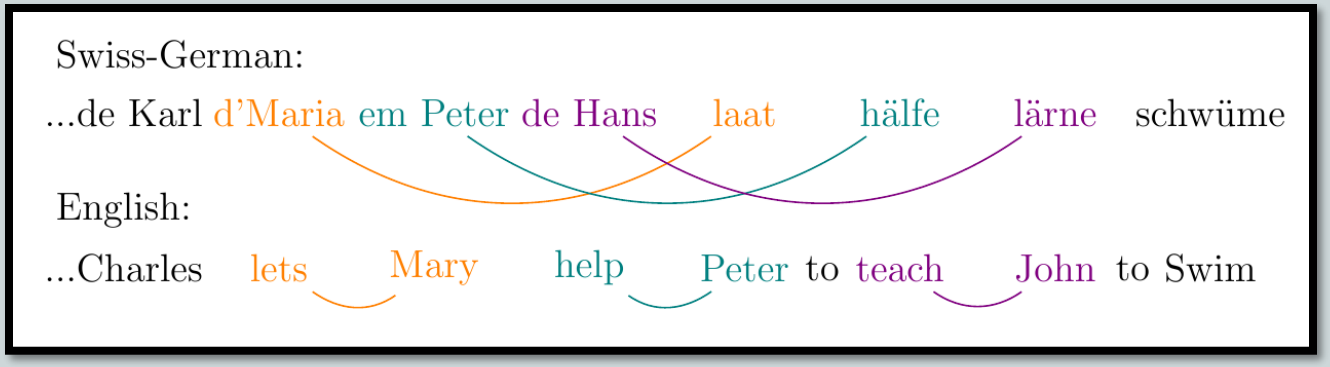
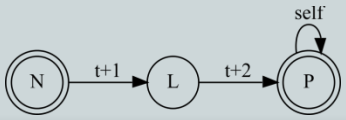
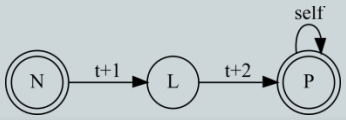


Image: wikimedia



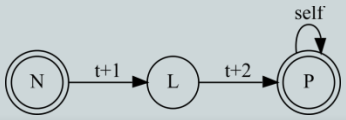
Context Free Grammars

- CFGs are nevertheless enough for most structures and much more efficient to compute
  - A context free grammar is a set of (de)composition rules over a set of symbols:
    - NP > DT NN
    - NP > NNP
    - DT > the
    - NN > house
    - NN > mouse
    - …
  - Symbols which do not decompose are called **terminals** (often =tokens)



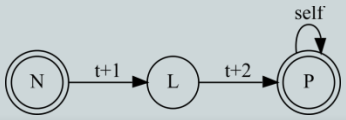
Context Free Grammars

- The set of decomposition combinations generates all utterances in the language **L** modelled by the grammar
- A **starting symbol** must be selected to generate from; usually S



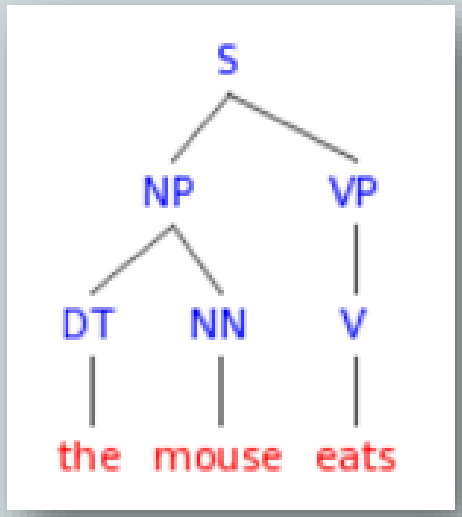
Context Free Grammars

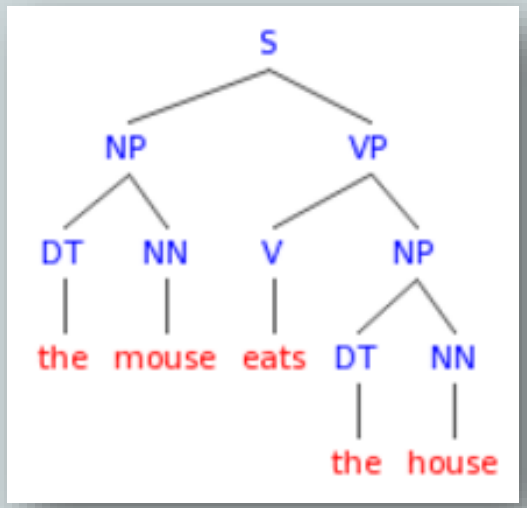
- Some example rules:
  - S > NP VP
  - VP > V NP
  - VP > V
  - V > eats
  - NP > DT NN
  - NN > mouse
  - NN > house
  - DT > the
  - …

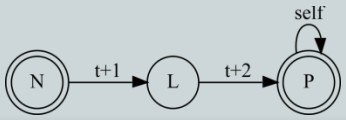


Now we can generate…

- (never minding meaning – à la 'colorless green ideas…')

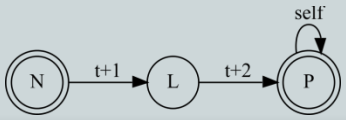






Exercise

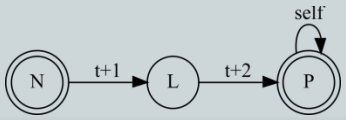
- Let's try to extract **context free rules** from sentences:
  - Every sentence has **S** at the top
  - Breaks down into phrases
  - Phrases decompose into our POS tags/other phrases
  - POS tags lead to tokens



Exercise

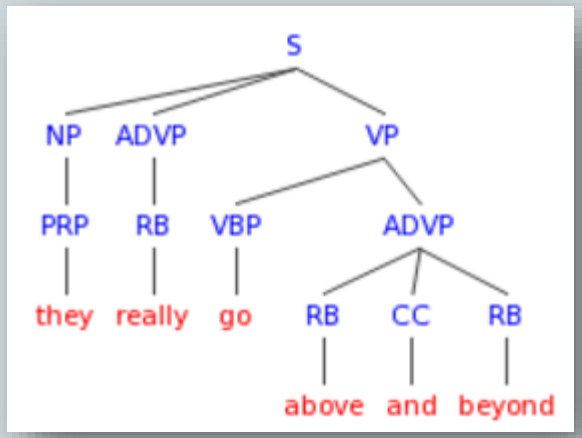
- Example:
  - They really go above and beyond!
- Tag it first:
  - PRP RB VBP RB CC RB .

  So we have **lexical** rules:
  - RB > really
  - VBP > go
  - …
- What are the **phrase structure** rules?

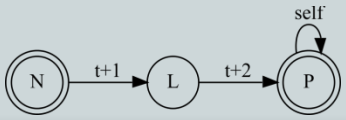


Exercise

- A possible analysis (English Web Treebank; other analyses are possible!)

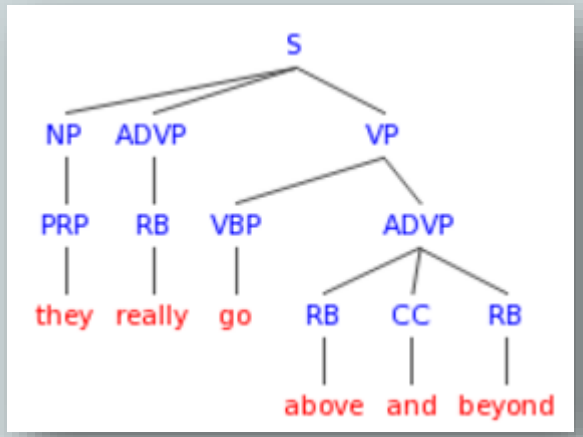


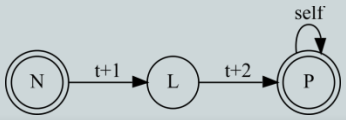
- How can we write the rules?



Exercise

- Break down the transitions:
  - S > NP ADVP VP
  - NP > PRP
  - ADVP > RB
  - VP > VBP ADVP
  - ADVP > RB CC RB

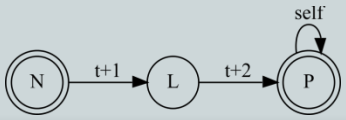




Your turn!

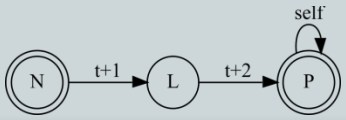
- Go to:
  - https://corpling.uis.georgetown.edu/etherpad/p/cfg

- Tag your assigned sentence
- Add transition rules for the tags to your rules
- Analyze syntax on paper
- Add transition rules for the phrases



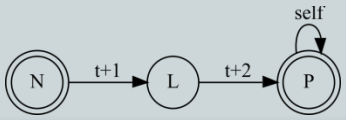
How do we prevent mistakes?

- Note that if we use traditional V for verbs:
  - S > NP VP
  - VP > V NP
  - NP > DT N
  - V > bite
  - N > dog
  - N > boy
- We can generate:
  - ***The dog bite the boy***
- Where is agreement in our grammar?



Add more categories

- You can now start to guess why these tags are important:
  - S >* … DT NN VB**Z** DT NN
  - S >* … You VB**P** DT NN

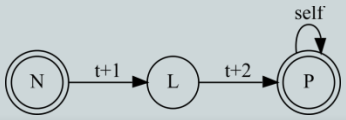


How do we prevent mistakes 2?

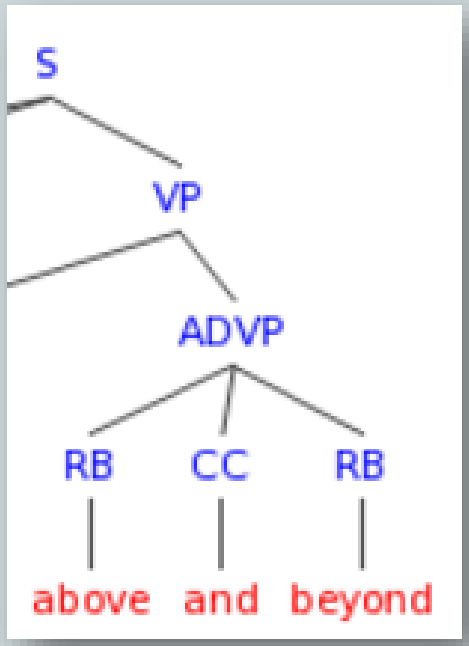
- What about specific participant patterns?
- We call these **subcategorization frames**:
  - * ***Jack slept the cake***
  - * ***Jill devoured***

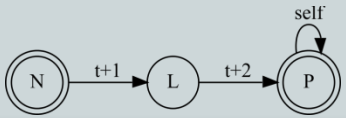
- We can make special categories:
  - VI > sleep
  - VT > devour

➢ In practice, this may not be needed (statistical parsing)



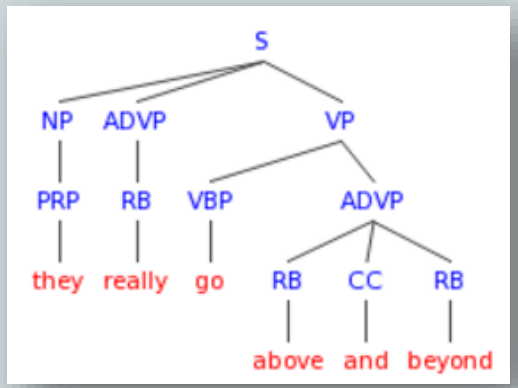
What's wrong with this picture?

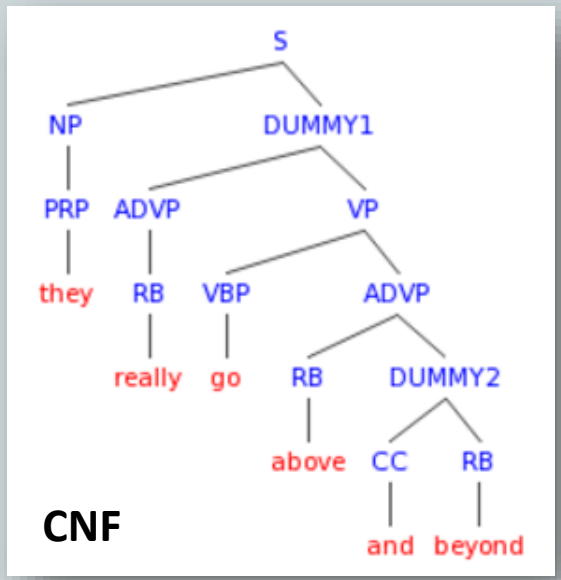


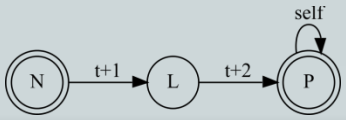


Aren't trees supposed to be binary?

- Actually, any *n*-ary tree can be turned into a binary tree **without loss of information**
  - Binarized trees are also called the **CNF** or **Chomsky Normal Form** of the tree
  - Ternary trees (or more) are a matter of taste in NLP





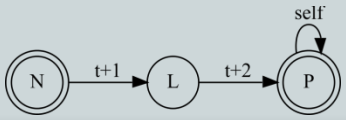


Context Free Grammar - Definition

- As a more formal definition, a CFG "G" is defined as:

G ≡

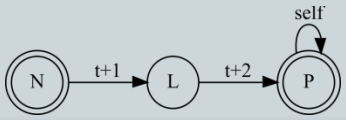
N	Set of **non-terminal** symbols
Σ	Set of **terminal** symbols (not in N!)
R	Set of rules of the form $A \rightarrow \beta$ where $A \in N$, $\beta \in (\Sigma \cup N)^*$
S	The designated start symbol



Great, but…

- Can we really build a grammar of English generating all possible sentences?
  - "All grammars leak" (Sapir 1921):
    - Carlson & Roeper (1980): prefixed verbs don't take PPs
    - *I want to overindulge in you* (Sampson 2007)
  - Not all "grammatical" structures are acceptable/occur:

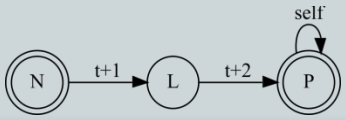
    Unlimited center embedding: S > NP S VP
    - x1 :*The boy the cat bit laughed*
    - x2: ?*The boy the cat the dog licked bit laughed*
    - x3: **The boy the cat the dog the girl bought licked bit laughed*
  - "Usage" effects, e.g. the above are better with pronouns:
    - *The boy the cat I bought licked laughed*



Great, but…

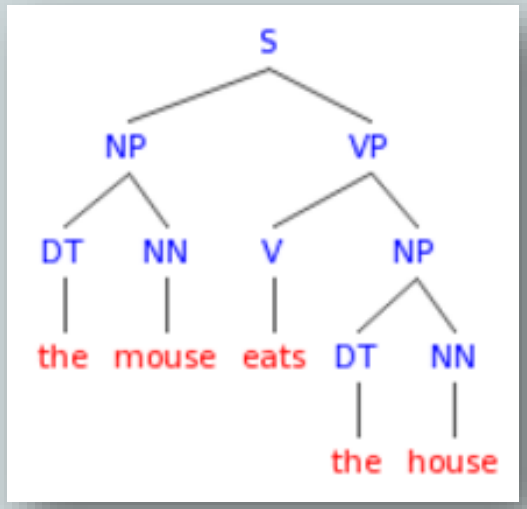
- Can we really build a grammar of English generating all possible sentences?
  - Much discussion about graded grammaticality (is every possible utterance really "in" or "out"?) – Sampson (2007)
  - Overgeneration is a serious issue – each new rule predicts a plethora of structures we will likely **never** see
  - Hand crafting a grammar beyond NP > DT NP becomes exponentially harder

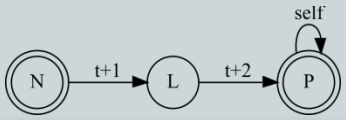
- Can we get a set of rules based on actual usage?



Yes, we can!

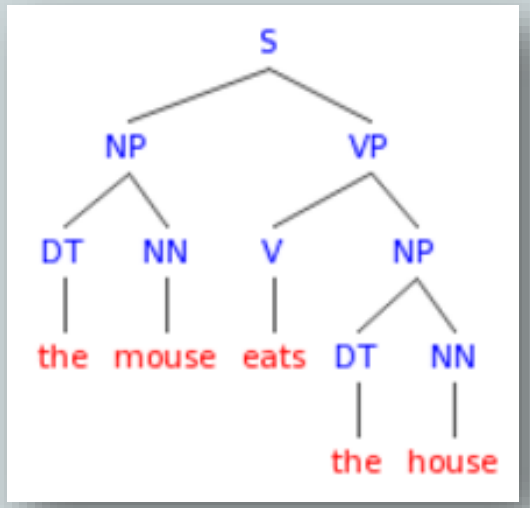
- Grammars can be induced from annotated data just like in our exercise
- In some ways, a corpus of syntax trees – a Treebank – **is** a grammar
- How often do we need the rules inherent in:

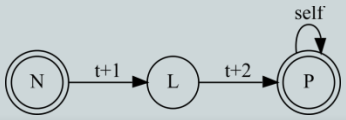




Answer

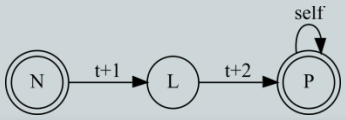
- S > NP VP (once)
- VP > V NP (once)
- NP > DT NN (twice)
- DT > the (twice)
- NN > mouse (once)
- NN > house (once)





Saving probabilities

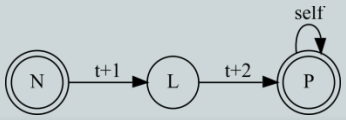
- It's easy to note how often each rule occurred,
  - Saving this data gives us an idea of how likely each decomposition is
  - Maybe we do need a rule for:
    - VP > V PP (overindulge in you)
    - V > over+V (overindulge)
  - But it's very rare/unlikely
- Saving the probabilities gives us a **Probabilistic Context Free Grammar** (PCFG)



How many rules?

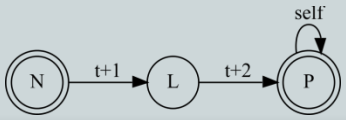
- Would we get a lot more rules than we could come up with by hand?
- The Penn Treebank Wall Street Journal corpus contains about 1 million tokens
- How many distinct **non-lexical** transition rules does it contain (incl. POS)?

- ~17,500 (Jurafsky & Martin 2008:408)



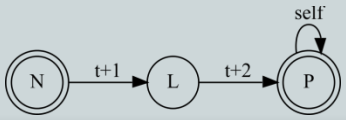
Very nice, but…

- So far we can only **generate**
  - Make all possible utterances using rules from a grammar or treebank
  - We could build a ‘tree-chatbot’ ☺
- In NL**P** we are less interested in generating
  - Random sentences are nice
  - But we want to **process** actual sentences generated by humans
  - Gateway to Natural Language **Understanding** (NLU)



How can we recognize a parse?

- Is this an English sentence?
  - ***The cat the dog the mouse licked bit ran***
- Two ways to check:
  - **Top down**, we generate all possible sentences:
    - S > NP S VP (twice)
    - S > NP VP (once)
    - …
    - V > ran
  - Did it appear in the list?



How can we recognize a parse?

- Is this an English sentence?
  - ***The cat the dog the mouse licked bit ran***
- Two ways to check:
  - **Bottom up**, we try to build phrases from words:
    - DT > The : means we might have a DT here
    - …
    - NP > DT NN : means we might have an NP
    - …
    - S > NP VP : OK, we've reached start symbol, all good!