Math 322

Ch 13 Modeling Computation

→ study human language → grammars → Computational Models
→ graph theory →

13.1 Languages and Grammars

① Natural Language (vs) Formal Language

② Syntax (form) (vs) Semantics (meaning)

→ Study : Formal Syntax

Math : toys + rules

toys : Sounds = encode ideas / emotions

Rules: How to construct complex ideas/emotions from simple ones.

↑ grammar

---

ex

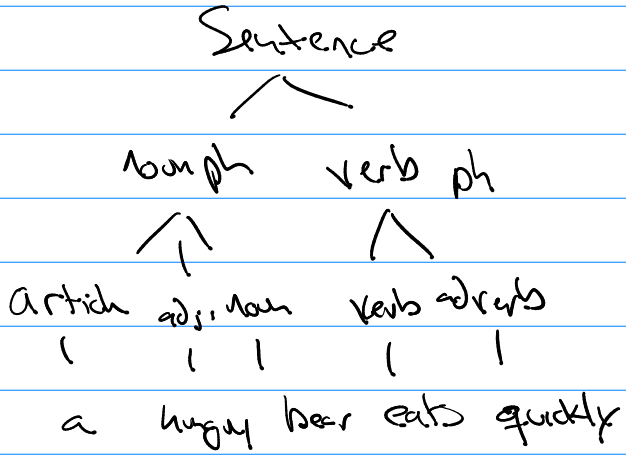
noun → rabbit or mathematician or bear
adjective → large or hungry or silly
article → the or a
verb → eats or hops or runs
adverb → wildly or quickly

Sentence → noun phrase , verb phrase

noun phrase → article noun or article adjective noun

verb phrase → verb or verb adverb

(ex) A hungry bear eats quickly.



So let's make this a Maths.

Def ① V is a finite nonempty set of elements (symbols) is a vocabulary (or alphabet)

② A sentence (or word) over V is a string of finite length of symbols of V.

2a) λ or null-string is the sentence (or word) of no length.

③ Set of all possible sentences (or words) over V is V^*.

④ Language over V is a subset of V^*.

Roster or list form of a language

Set builder?

$L = \{ O \mid$ logical system of why sentence is in language $\}$

↑ Sentence

Phrase-Structure Grammar $G = (V, T, S, P)$

① V is the vocabulary (or Alphabet)

② $T \subseteq V$ is the set of terminals
→ symbols that are non-replacement (specific low meaning)

③ $V - T = N$ set of non-terminals
→ symbols that are replaced

④ $S \in N$ is the start symbol.

⑤ P is a set of productions that tell you how to replace non-terminals.

left → right

ex) $G = (V, T, S, P)$

$V = \{a, 0, 1, q, A, B, S\}$

typical notation: $T = \{a, 0, 1, q\}$ lower case

$N = \{A, B, S\}$ upper case

S = Start symbol.

$P = \{ (S \to 0AB), (S \to q), (S \to A1),$
$(A \to 0A1), (AB \to BA01a),$
$(aA \to A a0) \}$

Derivations

ex $(aA_1 \to a001) \in P$

String: $01|a\boxed{A}1|0a \Rightarrow 01|a\boxed{001}|0a$ direct derivation

uses above production

Seq of strings: $w_0 \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_k$

$w_0 \overset{*}{\Rightarrow} w_k$ derivation.

ex $P = \{(S \to \lambda), (S \to AB), (A \to a), (A \to aBa), (B \to bb), (B \to Ab), (AB \to BA)\}$

$S \underset{2}{\Rightarrow} AB \underset{3}{\Rightarrow} BA \underset{4}{\Rightarrow} BaBa \underset{5}{\Rightarrow} bbaBa$

$\underset{6}{\Rightarrow} bbaAba \underset{3}{\Rightarrow} bbaaba$

$S \overset{*}{\Rightarrow} bbaAba$ not all terminals

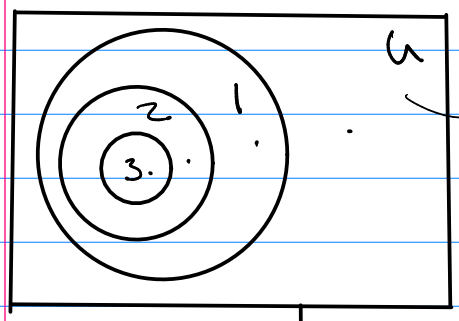
$S \overset{*}{\Rightarrow} bbaaba$ all terminals

Language of a grammar

$$L(G) = \{ \underset{\text{string}}{w} \in T^* \mid S \overset{*}{\Rightarrow} w \}$$

ex So $bbaaba \in L(G)$ above

Different forms (classes) of productions define different forms (classes) of grammars and L(G)



type 0

(left → right)

Restrictions on P

type 0	phrase-structure grammar	left must contain non-term
type 1	Context sensitive grammar	allow $S \to \Lambda$ all others are like $lAr \to lwr$ ↑ does not contract.
type 2	Context free grammar	left side must now only be single non-term. $A \to u$ $B \to v$ $S \to w$ $S \to \Lambda$
type 3	regular grammar	right side is single terminal or term with non-term $A \to 0$ $A \to 0B$ $S \to \Lambda$