Math 322

Languages → Graph theory → Science of Computation

Languages → Grammar

Language over Σ = Set of symbols (non-empty, finite)

is a subset Σ^*. **Note:** $|\Sigma^*| = \aleph_0$

countably infinite

$P(\Sigma^*)$ has all subsets as elements

So $P(\Sigma^*)$ is all possible languages..

$$|P(\Sigma^*)| = 2^{|\Sigma^*|} = 2^{\aleph_0} = \aleph_1$$

uncountably infinite

Define a Language

① **List** $L = \{$ set of things recognized under the Language $\}$

② **Build a Language?**

ⓐ build the strings of the Language.

ⓔⓧ Regular Expressions forming Regular Language

↳ Inductively defined

Basis: $\{\}$, $\{\varepsilon\}$, $\{a\}$ $a \in \Sigma$, are all regular expressions

Inductive: S_1, S_2 are regular expressions

i) $S_1 S_2$ is regular (concat.)

ii) $S_1 | S_2$ (is $S_1 \cup S_2$) is regular

iii) S_1^* is regular

So any language you could build using regular expressions is a regular language.

ⓔⓧ $\Sigma = \{0, 1\} = \{0 | 1\}$

$L = 0\,\boxed{\{0|1\}^*}\,11 = \{011, 0011, 0111, 00011, 00111, 01011, 01111, \ldots\}$

what about ? $L = \{01, 0011, 000111, 00001111, \dots\}$

$$L = \{0^n 1^n \mid n \geq 1\}$$

(non-regular)

---

(5) build the language of purposeful rules — productions

Phrase-Structure Grammar

(Note: we will talk about ch 4 ... backwards)

① Σ is still an non-empty set of symbols

② Σ^* is still set of all possible concatenations

③ $\Sigma = T \cup N$, disjoint sets

$T \equiv$ set of terminal symbols
$N \equiv$ set of non-terminal symbols

ex: English one of N is "Sentence"

Purpose is to be replaced.

④ $S \in N$, S is the start symbol

S

productions

Non-term

leaves = terminals

(5) a string is a finite length of symbols

(6) ε is the string of no length

(7) Languages are still a subset Σ^*

$\boxed{\text{but}}$ we want to build languages.

Productions: $\square \rightarrow \Delta$ rules of replacement

left string replaced by right string

(ex)

1. $S \rightarrow aA$
2. $A \rightarrow ab$
3. $A \rightarrow bB$
4. $B \rightarrow b$

Productions

$\Sigma = \{a, b, A, B, S\}$

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

$T = \{a, b\}$

start symbol $= S$

(ex) $\underbrace{S \underbrace{\boxed{\underset{(1)}{\Rightarrow}}}_{\text{directly derives}} aA \underset{(3)}{\Rightarrow} abB \underset{(3)}{\Rightarrow} abbB \underset{(4)}{\Rightarrow} abbb}_{\text{derivation}}$

$S \overset{*}{\Rightarrow} abbb$

Says abbb is derivable from S

Given a grammar G which is

$$G = (\Sigma, N, T, S, P)$$ productions

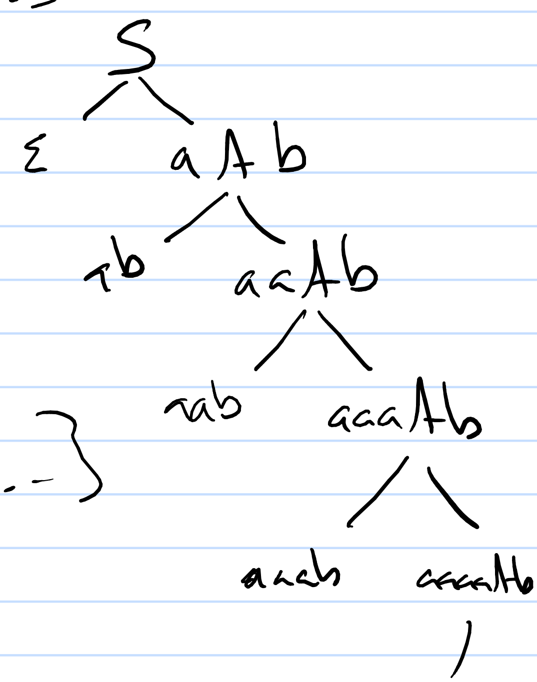
Language of G is

$$L(G) = \{ w \in T^* \mid S \xRightarrow{*} w \}$$

---

ex $N = \{S, A\}$ $T = \{a, b\}$

$$P = \left\{ \begin{array}{l} S \to \varepsilon \\ S \to aAb \\ A \to aA \\ A \to \varepsilon \end{array} \right\}$$

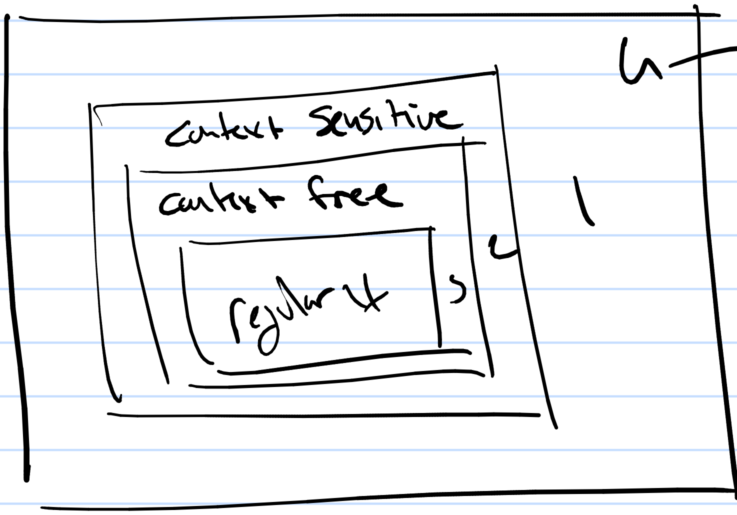


$L(G) = \{\varepsilon, ab, aab, aaab, aaaab, \ldots\}$

$L(G) = \{\varepsilon \mid a^n b \quad n \geq 1$

So, $L(G)$ are dependent on the $\boxed{\text{productions}}$

restrictions on productions will form families of $L(G)$.



→ all phrase structure grammars (no restrictions upon productions)

Name	Type	restrictions on P
phrase-structure grammar	0	none
context sensitive grammar (non-contracting)	1	$S \to \varepsilon$ $l A r \to l w r$ and $w \neq \varepsilon$
context free grammar	2	Now add restriction left $\to$ ↑ is always a single non-terminal
regular	3	$(S \to \varepsilon)$ is still ok $A \to aB$ or $A \to a$ all pro[ductions] have that form

$P = \{S \to \varepsilon,\ S \to aA,\ S \to bB,\ A \to aB,\ B \to b,\ A \to a\}$

type 0, 1, 2, $\boxed{3}$ Regular

$P = \{S \to \varepsilon,\ S \to Aa,\ S \to bB,\ A \to aB,\ B \to b,\ A \to a\}$

type 0, 1, 2 not 3

context free (not regular)