Math 322

$\boxed{Q's}$ 100 people out n, i, l

$i = 100$ $m = 4$

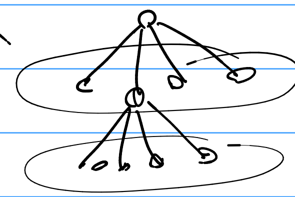
$n = 100 + l$
$n = 4 \cdot 100 + 1$ } $l = 301$, $n = 401$

$h \geq \lceil \log_4 301 \rceil$

$\underline{h \geq 5}$

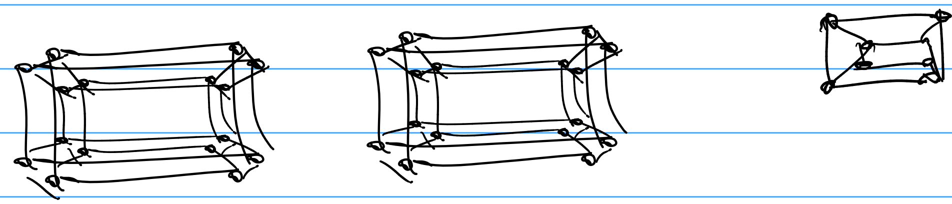
$n = i + l$

$n - i = l$



$\log_m (l)$

Q_5



$\deg(x \in Q_n) = n$

---

13 Computational Machines (Turing Machines)

↑ sets & 5-tuples

13.1 languages / grammars

$\boxed{\text{Language}}$ ① Natural language

My idea $\xrightarrow{\text{language}}$ your idea

"recognize"

② ~~Natural~~ vs formal

syntax, ~~semantics~~

↑

ex "English"

p_1: Sentence → noun phrase, verb phrase
p_2: noun phrase → article, adjective, noun

non-terminals

p_3: noun → rabbit or dog or house
p_{8}: adjectives → large or small or happy
$p_{9,10}$: article → a or the

terminals

p_{11}: Verb phrase → Verb, adverb
verb → eats or hops or hits
adverb → quickly or wildly

terminals

Natural Language: does the collection of ideas form recognition?

Formal: use phrase-structure grammar

1. V a non-empty set of elements (symbols): Vocabulary (Alphabet)
2. sentence (word) is a string over V of finite length.
3. λ: empty string of no length.
4. V^*: set of all sentences (words) over V.

5. Language is a subset of V^*.

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

V: Vocabulary (non-empty)
V = terminal symbols + non-terminal symbols
$V = T \cup N$ $(N = V - T)$

$S \in V$, S is the start symbol (non-terminal)

(ex) above we use S = Sentence

P is the set of productions. Each production tells you how to replace non-terminals by (non-term + terminals)

$P = \{P_1, P_2, \dots, P_k\}$

P_i is left side string $\rightarrow$ New string

(ex) Verb phrase $\rightarrow$ Verb, adverb $= P_1$
verb $\rightarrow$ eats or hops or hits $= \boxed{P_2}, \boxed{P_3}, P_4$
adverb $\rightarrow$ quickly or wildly

Verb $\rightarrow$ eats

verb $\rightarrow$ hops

using productions: $P_i : z_0 \rightarrow z_1$

given word $l z_0 r \Rightarrow l z_1 r$ new word

$$w_0 \underset{P_i}{\Rightarrow} w_1$$

using productions is a derivation.

(ex) $w_0 \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_n$

If a derivation exists $w_0 \overset{*}{\Rightarrow} w_n$, w_n derives from w_0

Def language generated by a grammar $L(G)$

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

ex

non-terminals

$V = \{a, b, c, A, B, S\}$

terminals ... start symbol

$P = \{ \underbrace{aAb \to acbb}_{1},\ \underbrace{S \to AB}_{2},\ \underbrace{A \to aA}_{3},\ \underbrace{B \to b}_{4} \}$

Some derivations: $S \underset{2}{\Rightarrow} AB \underset{3}{\Rightarrow} aAB \underset{3}{\Rightarrow} aaAB \underset{4}{\Rightarrow} aaAb \underset{1}{\Rightarrow} aacbb$

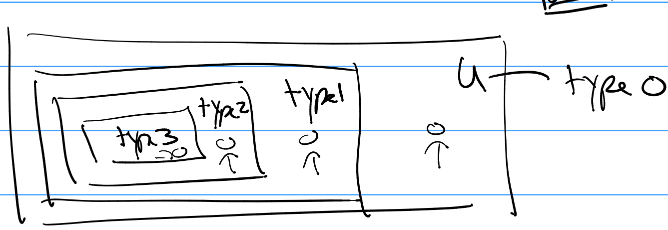
so $aacbb \in L(G)$

---

because productions create the $w \in T^*$ of a language then restrictions on productions will restrict the languages.

type	name	restrictions	Note:
0	phrase-structure grammar	none	(all terms ⇒ ??) can not be fully replaced.
1	context-sensitive	$p_i: lAr \to lwr$, $w \neq \lambda$; but $S \to \lambda$ is ok if S is never on the right.	
2	context-free	(all restricts from 1) plus left side is a single non-term.	
3	regular	(all restrictions from 1 and 2) plus right side is a or aB	

Note:



ex $P = \{ S \to \lambda, S \to ABc, AB \to ab, A \to a, B \to c \}$

not type2

type 0, $\boxed{\text{type 1, not type 2}}$

ex $P = \{ S \to \lambda, S \to \boxed{aA}, A \to \boxed{a}, A \to \boxed{aA}, A \to \boxed{Ab} \}$

not 3

type 0, type 1, $\boxed{\text{type 2, not type 3}}$

$$S \underset{2}{\Rightarrow} aA \underset{4}{\Rightarrow} aaA \underset{5}{\Rightarrow} aaAb \underset{5}{\Rightarrow} aaAbb \underset{5}{\Rightarrow} aaAbbb \underset{3}{\Rightarrow} \underline{aaabbb}$$