Unit-2 Formal Languages

Monday, May 3, 2021 9:51 AM

Lets take an example of English language sentence

"Ram ate quickly"
<noun><verb><adverb>

so $S \rightarrow$ <noun><verb><adverb>

ex "Sam ran"
<noun><verb>

$\Rightarrow$

<noun> → Ram / Sam
<verb> → ate / ran
<adverb> → quickly

Variables — terminals

* So, we can frame a grammar of any language to write the sentence correctly.

Definition of Grammar : (Noam chomsky)

A Phrase-structure grammar $\Rightarrow \langle V_N, \Sigma, P, S \rangle$

where,

$V_N \Rightarrow$ finite non-empty set of variables → $A, B, C, A_1, A_2 \cdots$

$\Sigma \Rightarrow$ finite non-empty set of terminals → $a, b, c,$

$P \Rightarrow$ set of production rules

$x, y, z, w \rightarrow$ strings of terminals

ex $V_N = \{$<noun> <verb> <adverb>$\}$

ex $\Sigma = \{$ Ram, Sam, ate, ran, quickly $\}$

ex
$S \rightarrow$ <noun><verb><adverb>
$S \rightarrow$ <noun><verb>
<noun> → Ram / Sam ...
<verb> → ate / ran
<adverb> → quickly / ----

$S \rightarrow$ Ram ate quickly

* Elements of P are $\alpha \rightarrow \beta$, where α & β are strings on $V_N \cup \Sigma$

$S \Rightarrow$ special variable ($\in V_N$) called as start symbol.

* Can we construct a Finite Automata which accepts any language or for any given language is it possible to construct a FA ? $\Rightarrow$ limitation of FA

v. imp

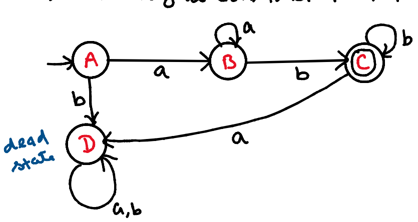
ex $L = \{a^n b^n \mid n \geq 1\} \rightarrow$ infinite

$L = \{ab, aabb, aaabbb, \cdots\cdots\}$ infinite

$\Sigma = \{a, b\}^*$

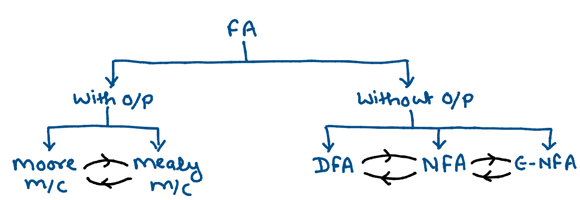
here, to design a FA which can accept this type of language, the m/c (FA) will require an infinite memory.

so, lets try to construct a FA for such lang.



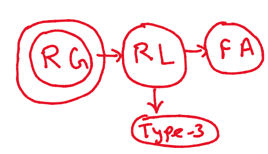
Observations:

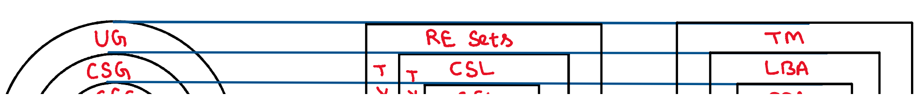
* This m/c can accept all the strings of given lang.
* It will accept the strings such as aab or abb which are not given in the language → limitation of FA
* Only we are able to check that whether all b's are followed by a's or not.
* we can not keep a count of number of a's and compare it against no. of b's.
* In order to count no. of a's corresponding to b's an infinite memory along with FA is required.
* But FA doesn't have infinite amount of memory
* So, we need next level m/c
* What we have studied so far

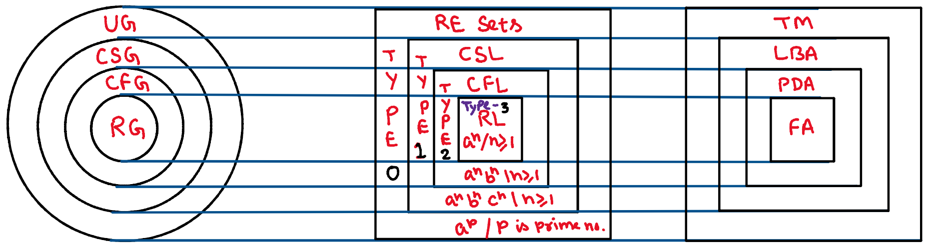


FA + memory $\Rightarrow$ PDA (Push down Automata)

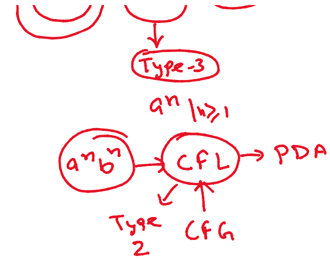
Family of Grammar / Languages / Machines :







family of grammars   family of languages   family of m/cs



Abbreviations:

UG → unrestricted grammar
✓ CSG → Context sensitive grammar
✓ CFG → Context free grammar
RG → Regular grammar

RE → Recursively enumerable Sets
CSL → Context Sensitive Language
CFL → Context Free Language
RL → Regular Language

TM → Turing Machine
LBA → Linear Bounded Automata
PDA → Push down Automata
FA → Finite Automata