Math 322

5)

$S \to \lambda$
$A \to a$
$A \to aB$

regular set

generated by regular grammar

recognized by FSA

Grammar types

type:	production
0	no – restrictions
1	non-contraction outside of $S \to \lambda$
2	context free: all productions left → right (single non-terminal)
3	regular: $S \to A$, $A \to a$, $A \to aB$

type 0 all phrase structure grammars

type 1 context sensitive

type 2 context free

type 3 regular (FSA)

ex

$A \to BcAcc$
$S \to \lambda$
$B \to cb$
$A \to a$

Because FSA recognition ≡ produced by regular grammar

7) $S \to \lambda$, $S \to 0A$, $A \to AB$, $A \to 01$, $B = 00$

type
0 always true
1 true
2 true
3 no

→ type 2 (not 3)
context-free not regular.

ex is a context free language, ~~not~~ regular

$L(G) = \underline{\lambda, 001(00)^*}$

$\underline{S \to 0A}$, $\underbrace{A \to AB}_{P_2}$, $\underbrace{A \to 01}_{P_3}$, $\underbrace{B = 00}_{P_4}$

$S \to 0A \xrightarrow{P_3} \boxed{001}$

$0A \xrightarrow{P_2} 0AB \xrightarrow{P_3} 001B \to 00100$

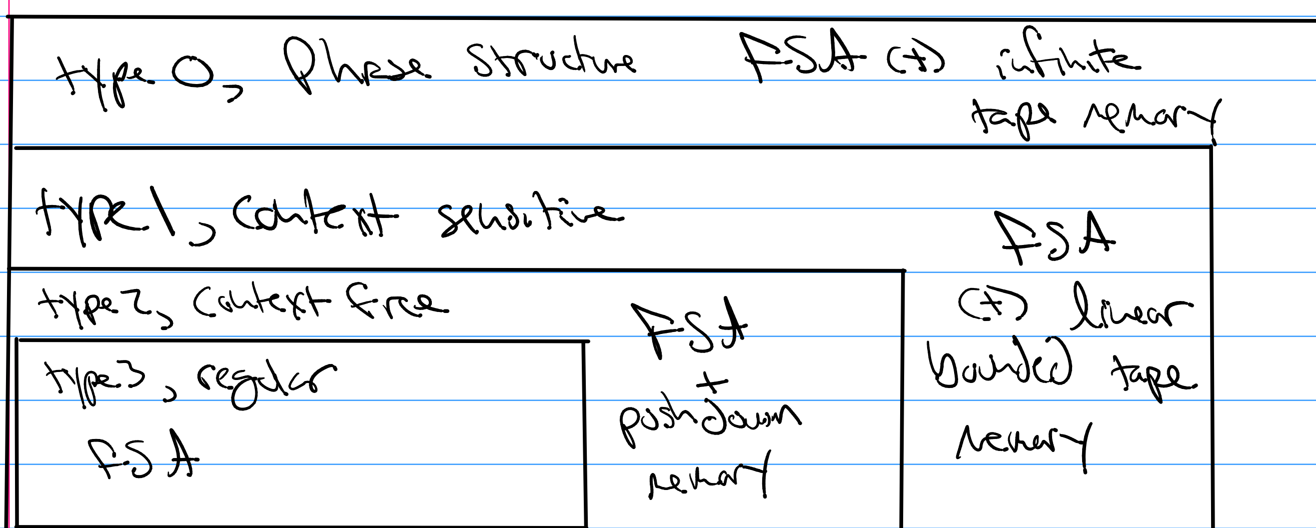
$0AB \xrightarrow{P_2} 0ABB \xrightarrow{P_{3,4}} 0010000$

$0ABB \to 0ABBB$

So $L = \lambda, 001(00)^*$ is not regular

$\equiv$ no FSA that recognizes it.

More power machines



$\boxed{\text{th}^m}$ context free language iff push-down machine recognizes it.

$\boxed{\text{th}^m}$ context-sensitive language iff linear bounded machine recognizes it.

Thm any phrase structure language iff FSA with infinite tape memory. → Turing Machine

---

Turing Machines:

$$T = (S, I, f, s_0)$$

S, set of states

I, input alphabet (must have, B, the blank symbol)

f a partial function

$$f: S \times I \to S \times I \times \underbrace{\{\text{left}, \text{right}\}}_{\text{motion}}$$

(Note: f is a set of 5-tuples)

s_0 is the start state.

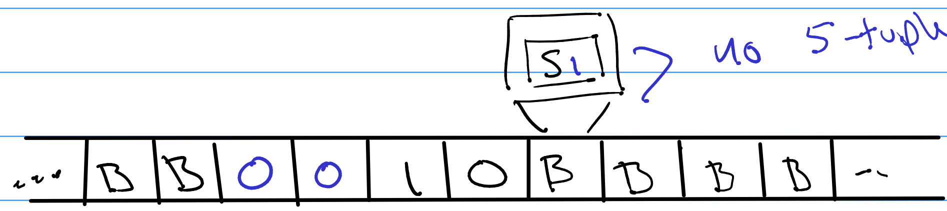
Final States? are any state in S that are not the 1st element of a 5-tuple of f.

Ex T,

$S = \{s_0, s_1, s_2\}$ → Final states $= \{s_2\}$ b/c not a 1st element of f

$I = \{B, 0, 1\}$

$f = \{ (s_0, 0, s_0, 0, R), (s_0, 1, s_1, 0, R),$
$(s_1, 0, s_1, 0, R), (s_1, 1, s_1, 1, R) \}$



no 5-tuple with $(s_1, B, \ldots)$

Halt

so, T halted in S_1 a non-final state.

To study/create Turing machines to figure out 5-tuples to do tasks on the tape.

Number Theoretic Functions. $1+2=3$

task: add non-neg integers.

input: Unary system
$0 = 1$
$1 = 11$
$2 = 111$
$3 = 1111$

operator symbol, $*$

$1+2 \rightarrow$ on tape --BB11 * 111 BB--

goal   5-tuples?

$3 \rightarrow$ halt --B1111B--
(final state
