Math 322

FSA

FSA

regular set

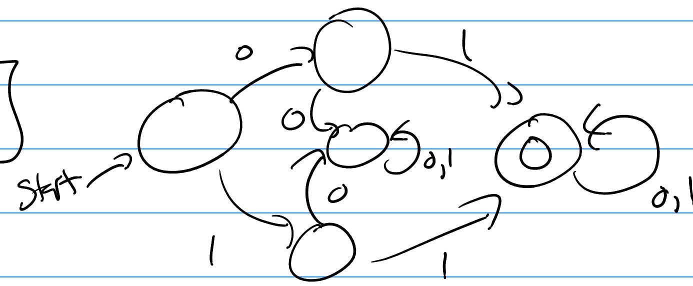
regular grammar

ex $(0\cup 1)1(0\cup 1)^*$

$M = ?$

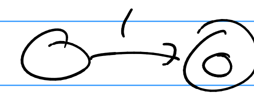
$(0\cup 1)1(0\cup 1)^*$

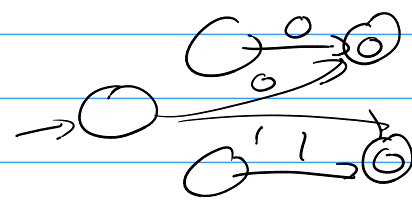
create FSA



constructive way

M_0 : 

M_1 : 

$M_{0\cup 1}$ 

etc (finish @ home)

What about $1^n 0^n = \{\lambda, 10, 1100, 111000, 11110000, \ldots\}$

→ context free set $\neq$ FSA

$\neq$ regular grammar

but

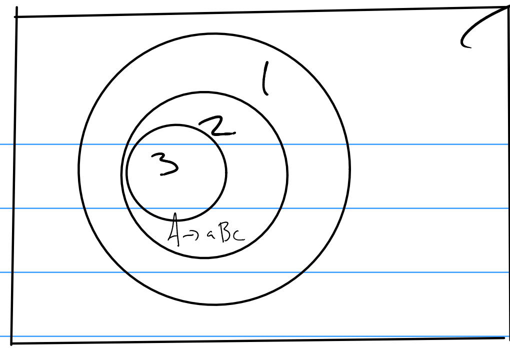
context free set

Machine?

context free grammar

What machine can recognize context free grammars?

Grammars



type 0
phrase structure grammar

$A \to aBc$

$aA \to aB$

$A \to a$

type 3 ⊂ type 2 ⊂ type 1 ⊂ type 0

Machines: FSA ⊂ pushdown automata → F.S.A with pushdown stack memory.

regular — context free

FSA ⊂ pushdown automata ⊂ linear bounded automata → FSA with linear size tape memory.

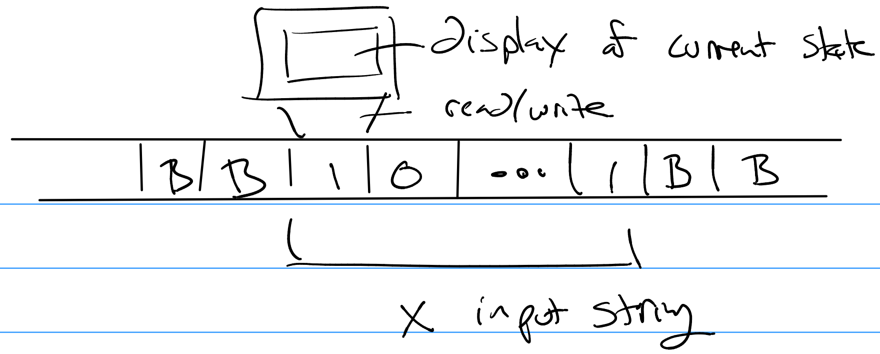
context sensitive

FSA ⊂ pushdown ⊂ linear bounded ⊂ Turing Machine — FSA with infinite tape memory

regular — context free — context sensitive — any phrase structure grammar.

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

- S : States
- I : Input Alphabet (must have a blank symbol, B)
- s_0 : Start State
- $f : S \times I \to S \times I \times \{\text{left}, \text{right}\}$

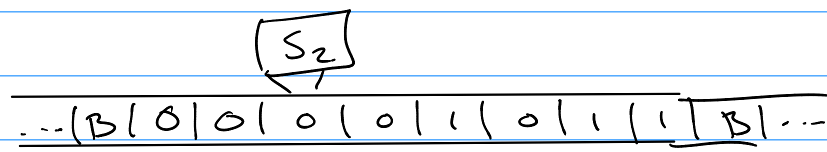


$\Gamma = \{0, 1, B\}$

Give 5 tuples (Input pair, output triple) to represent T.

ex $x = 11001011$

$$T = \{(s_0, 1, s_1, 0, R),\ (s_0, 0, s_0, 0, R),\ (s_1, 0, s_1, 0, R),\ (s_1, 1, s_2, 0, R),\ (s_2, B, s_2, B, R)\}$$



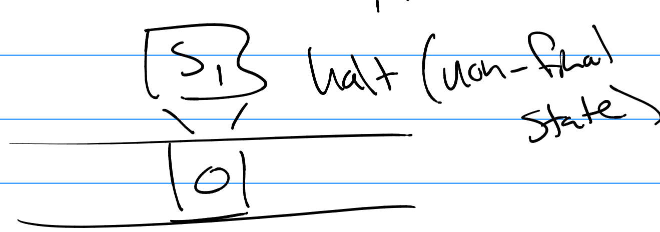
Final States? → are the states that never show up as a state in an input pair.

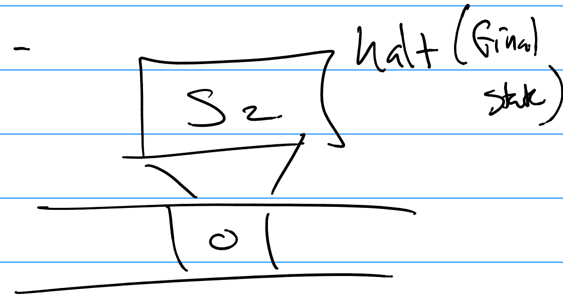
ex $S = \{s_0, s_1, s_2, s_3\}$

$$T = \{(s_0, 0, s_1, 0, R),\ (s_0, 1, s_0, 0, R),\ (s_1, 1, s_2, 1, L)\}$$

→ s_2, s_3 are final states

Now: we have two types of halts ..





→ Compute Number Theoretic Functions. (unary)

$0 = 1$

$1 = 11$

$2 = 111$

$3 = 1111$

$\vdots$

11 * 11

111

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

$3 + 2 = \overbrace{\boxed{1111} * \boxed{111}}^{3 \;+\; 2}$ input

T = 5 tuples?

$\boxed{111111}$ output