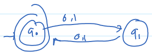
Friday, June 2, 2023 12:59 PM

1) L is decidable if there is a TM M that ALWAYS HALTS and on input x
   1) M halts in accept if $x \in L$
   2) M halts in reject if $x \notin L$

2) Machine descriptions can be coded as a string

Ex: DFA



can be described as: $\underbrace{10}_{\#\text{ of states}}$ # $\underbrace{0}_{\text{accepting states}}$ #0,0,1;0,1,1;1,0,0;1,1,0;#

NOTATION: ⟨M⟩ denotes description of M

We can encode PDA, CFG, regular expressions as strings also.

Languages about descriptions of DFA

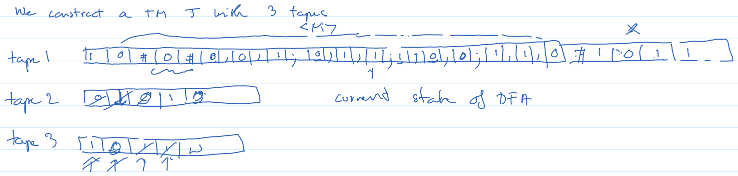
$A_{DFA} = \{ \langle M \rangle \# x : \text{DFA } M \text{ accepts } x \}$

Claim: A_{DFA} is decidable.

Proof:

We construct a TM J that always halts such that given ⟨M⟩#x as input
   1) J halts in accept if DFA M accepts x
   2) J halts in reject if DFA M rejects x

We construct a TM J with 3 tapes



Tape 1: ⟨M⟩#x

tape 2: current state of M q

tape 3: input x to M a

In each step, J searches for the transition (q, a, p) and updates state to p and move head 3 to the right.

When the 3^{rd} head reads ⊔, check to see if M's state is accepting state.

If yes, accept

If no, reject

$A_{NFA}: \{ \langle M \rangle \# x : \text{NFA } M \text{ accepts } x \}$

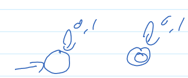
Claim: A_{NFA} is decidable

Proof: TM J converts NFA M to equivalent DFA N and checks if $\langle N \rangle \# x \in A_{DFA}$

$A_{REG} = \{ \langle R \rangle \# x : \text{reg exp } R \text{ generates } x \}$

Claim: A_{Reg} is decidable

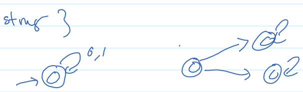
$E_{DFA} = \{ \langle M \rangle : \text{DFA } M \text{ accepts no strings} \}$



Claim: E_{DFA} is decidable

Proof: Given ⟨M⟩ construct the state diagram and perform a BFS at the start state. If no accepting states can be reached, accept; else reject

$U_{DFA} = \{ \langle M \rangle : \text{DFA } M \text{ accepts any string} \}$

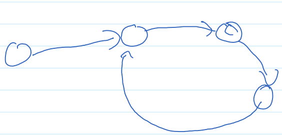


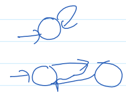
Claim: U_{DFA} is decidable

Given ⟨M⟩, construct ⟨N⟩ by flipping accepting/nonaccepting states

$\langle M \rangle \in U_{DFA} \iff \langle N \rangle \in E_{DFA}$

$F_{DFA} = \{ \langle M \rangle : \text{DFA } M \text{ accepts a finite number of strings} \}$





$I_{DFA} = \{ \langle M \rangle : \text{DFA } M \text{ accepts as many strings} \}$

Claim: M has q states

$L(M)$ is finite $\iff$ M accepts no strings of length between q and 2q

Claim: M has q states

$L(M)$ is finite $\iff$ M accepts no strings of length between q and $2q$

Proof: Pumping lemma.

$z = u\cancel{v}w$

$EQ_{DFA} = \{ \langle M \rangle \# \langle N \rangle :$ DFA M, N accept the same language $\}$

minimize M and N and check if the results are the same

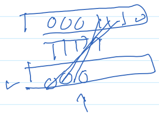
$A_{CFG} = \{ \langle G \rangle \# x :$ CFG G generates $x \}$

Decidable: on $\langle G \rangle \# x$, convert G to CNF

run CYK on G, x

$A_{PDA} = \{ \langle M \rangle \# x :$ PDA M accepts $x \}$

$U_{CFG} = \{ \langle G \rangle :$ CFG G generates all strings $\}$



1
3