5/28/2010

ANNOUNCEMENT: FINAL WILL BE A TAKE-HOME FINAL,
GIVEN ON FRIDAY AND DUE BY END OF DAY MONDAY
(AND GREAT HAPPINESS ENSUED)

Q. Does a DFA D accept an input w?
CAN you write a program to answer this question? Yes.
Can you make a Turing Machine answer this question? How?
ENCode D and w as input to a TM (strings)

INPUT STRING = $\langle D, w\rangle$

$$D = (Q, \Sigma, \delta, q_0, F)$$

$Q = \{q_0, q_1, \ldots, \}$, $\Sigma = \{0,1\}$

δ: $q_0, 0 \to q_{17}$; $q_{14}, 1 \to q_7$

$\Rightarrow$ FORMAT ALL THIS INFO INTO A SINGE STRING, D

$$D = \overset{Q}{\# q_0, q_1, \ldots} \; \overset{\Sigma}{\# 0, 1} \; \overset{\delta}{\# q_0, 0, q_{17} \$ q_{14}, 1, q_7 \$ \ldots}$$

$$\ldots \# \, q_0 \, \overset{F}{\# \, q_1, q_4, q_9} \, \overset{w}{\# \, 0110}$$

<u>THM 1</u> Language $A_{DFA} = \{ \langle D, w\rangle \mid D \text{ is a DFA that accepts } w\}$
is decidable

<u>PF.</u> CONSIDER DECIDER TM M_1 =

<u>LEVELS OF DESCRIPTION OF TMs</u>

1. Formal: $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{ACC}, q_{REJ})$

2. Implementation Level: Describe TM in English: tape, etc.

3. High level: algorithm (we can do this because the Church-Turing thesis tells us that all algorithms can be implemented as TM's)

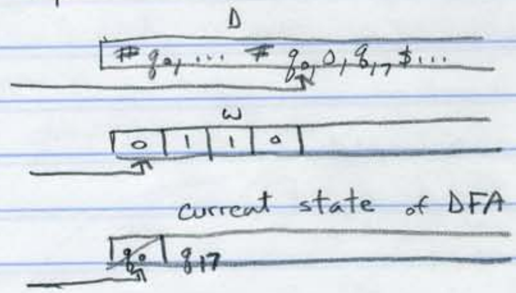
A_{DFA} is Turing-Decidable:

Pf. Consider Decider TM M_1 =

"ON INPUT s:

1. Check that $s = \langle D, w \rangle$
   (if not, reject s)
2. Simulate D on w

IDEA: 3 tapes:



Loop:

Read the state, read the input, read the transition, write the new state

At end of input:

Check current state to see if it's an accept state. Accept if yes, Reject if no.

A_{NFA} is Turing-Decidable:

$A_{NFA} = \{ \langle N, w \rangle \mid \text{NFA } N \text{ accepts } w \}$

M_2: (Convert N to DFA D_N and run M_1 on $\langle D_N, w \rangle$)

$A_{REX} = \{ \langle R, w \rangle \mid \text{Reg exp } R \text{ generates } w \}$

(Convert R to equivalent NFA N_R AND RUN M_2 on $\langle N_R, w \rangle$)

A_{REX} is Turing-Decidable

A_{CFG} is Turing-decidable:

$A_{CFG} = \{ \langle G, w \rangle \mid$ CFG G generates $w \}$

(convert G to CHOMSKY NORMAL FORM:
w is generated in $2|w|-1$ steps)
Now bound the machine: if w not generated within $2|w|-1$ steps → reject
(otherwise risk infinitely looping TM)

$A_{TM} = \{ \langle M, w \rangle \mid$ TM M accepts $w \}$

Is A_{TM} Turing-Decidable? NO!

[See slides]

Reducibility and the Halting Problem

[see slides]