

Chapter 4 - Decidability.

We know what an algorithm is - it is a TM.

TM deciders - on all inputs, halt and either ACCEPT.
or REJECT.

TM recognizer - on all $w \in L$, halt and ACCEPT.
on all $w \notin L$, does not accept. $\left\{ \begin{array}{l} \text{REJECT} \\ \text{or} \\ \text{loop} \end{array} \right.$

Language

$A_{\text{DFA}} = \{ \langle B, w \rangle \mid B \text{ is a DFA that accepts input string } w \}$

equivalent

eg $\langle (\{q_0, q_1\}, \{a, b\}, q_0, \{(q_0, a, q_1), (q_1, b, q_0)\}, \{q_1\}), aab \rangle$



Decision Problem:

Given: a DFA B , a string w

Decide: Does B accept w ?

Theorem 4.1 A_{DFA} is decidable (is a decidable language)

Proof: We propose that the following TM M decides A_{DFA} .

$M =$ "on input $\langle B, w \rangle$, where B is a DFA, w is a string:

1. Simulate B on input w .

2. If B accepts, ACCEPT.

If B ends in a non-accepting state, REJECT."

Why this works:

- We already know that a TM can simulate another TM.
- A DFA is a restricted form of TM:
(a TM that only moves head R, and does so once per transition; and it always halts when it reaches first \perp .)

◦ M can indeed simulate B,
and B always halts on all inputs,

◦ M halts on all inputs.

And M accepts $\langle B, w \rangle$ exactly when $w \in L(B)$. \square

$$A_{NFA} = \{ \langle B, w \rangle \mid B \text{ is an NFA that accepts } w \}$$

Theorem 4.2 A_{NFA} is decidable.

Proof: A TM that decides A_{NFA} is N , where:

$N =$ "on input $\langle B, w \rangle$ where B is an NFA and w a string:

1. Convert NFA B into DFA B' using the construction (algorithm) we covered in Thm 1.39
2. Run the TM M from Thm 4.1 on input $\langle B', w \rangle$.

2.1 if M accepts, ACCEPT.
if M rejects, REJECT. "

Step 1 is doable, because the construction has a finite # of steps.

Step 2 is doable because a TM can run another TM as a subroutine

And N is a decider, because M is a decider. \square

$$A_{\text{REX}} = \{ \langle R, w \rangle \mid R \text{ is a reg expression that generates } w \}$$

Theorem 4.3 A_{REX} is decidable.

Proof: The TM P decides A_{REX} .

$P =$ " on input $\langle R, w \rangle$, where R is a reg. exp and w is a string:

1. Convert R into a NFA X using the construction (alg) given in Thm 1.54.
2. Run N (the A_{NFA} decider) on $\langle X, w \rangle$
3. If N accepts, ACCEPT.
If N rejects, REJECT. "

Since N is a decider, so is P .

P correctly decides A_{REX} . \square
