Chapter 5 – Decidability.

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

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

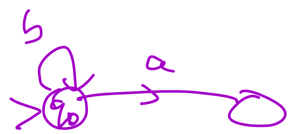
TM recognizer – on all $w \in L$, halt and ACCEPT.
on all $w \notin L$, does not accept. { REJECT or loop

Language

equivalent

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

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



Decision Problem:

Given: a DFA B, a string w

Decide: Does B accept w?

Theorem 5.1.2: A_{DFA} is decidable { i.e. the language A_{DFA} is decidable. I.e. the decision problem can be answered Yes/N

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 ⊔.)

∴ M can indeed simulate B on w
and B always halts on all inputs,
∴ M halts on all inputs.

Note that M accepts ⟨B, w⟩ exactly when $w \in L(B)$. ▨

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

Theorem 5.1.3 A_{NFA} is decidable.

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

N = "on input ⟨B, w⟩ where B is an NFA and w a string:

1. Convert NFA B into DFA B′ using the construction (algorithm) we covered in week 2.
2. Run the TM M from Thm 4.1 on input ⟨B′, w⟩.

   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. $\blacksquare$

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

Theorem 4.3 A_{REX} is decidable.

Proof: The TM 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.

$\therefore$ P correctly decides A_{REX}. $\blacksquare$