HW5 Problem 2 NOT on quiz!

HW 5 P3

$S = \{ \langle M \rangle \mid M$ is a DFA that accepts w^R whenever it accepts w. $\}$

Show S is decidable.

Sketch

- Construct M_R: M with flipped start + accept states
- Construct M^C: M with flipped accept states
  → $L(M) \cap L(M^C) = \emptyset$

$L(M_R)$ is going to be $w^R \ \forall w \in L(M)$

→ $L(M_R) \subseteq L(M)$ because M accepts all reverse w^R

$L(M_R) \cap L(M^C) = \emptyset$

HW5 Problem 2 NOT on Quiz!

HW5 Q2

$S = \{ \langle M \rangle \mid M$ is a DFA that accepts w^R whenever it accepts w. $\}$

Show S is decidable.

Solution:

- Construct M_R: M with flipped start & accept states
- Construct M^C: M with flipped accept states

→ $L(M) \cap L(\overline{M_R}) = \emptyset$

$L(M_R)$ is going to be w^R if $w \in L(M)$

→ $L(M_R) \cong L(M)$ because M accepts all reverse w^R

$L(M_R) \cap L(M^C) = \emptyset$

R = "On input $\langle M \rangle$ where M is a DFA,

1. Construct M_R:
   - construct $\overset{\text{NFA}}{N_R}$, which M with all start + accept states flipped + δ reversed
   - convert N_R to a ~~NFA~~ DFA M_R
2. Construct M^c:
   - flip accept states of M
   - (If $L(M^c) \cap L(M_R) = \emptyset$, then M <u>does</u> accept all reverse strings.)
3. Use E_{DFA} to see if $L(M^c) \cap L(M_R)$ is empty
   - → If empty, accept
   - → Else, reject. "

R' = "On input $\langle M \rangle$ where M is a DFA,

1. Construct M_R

   (Same as above)
2. Use EQ_{DFA} to see if $L(M) = L(M_R)$
   - → If yes, accept
   - → Else, reject "

R = "On input $\langle M \rangle$ where M is a DFA,

1. Construct M_R:
   - Construct M_R^{NFA}, which is M with all start & accept states flipped & arrows reversed
   - Convert N_R to a DFA M_R
2. Construct M^c:
   - Flip accept states of M
   - If $L(M^c) \cap L(M_R) = \emptyset$, then M does accept all reverse strings
3. Use E_{DFA} to see if $L(M^c) \cap L(M_R)$ is empty
   → If empty, accept
   → Else, reject."

R' = "On input $\langle M \rangle$ where M is a DFA

1. Construct M_R
   (Same as above)
2. Use EQ_{DFA} to see if $L(M) = L(M_R)$
   → If yes, accept
   → Else, reject"

Reducibility $(A \leq B)$

- Assume B is decidable
- Build machine that uses B to decide A

Mapping Reducibility $(A \leq_R B)$

- Build a machine M' that takes input x and returns y s.t.

$$x \in A \longleftrightarrow y \in B$$

$$\uparrow$$

$$f(x)$$

<u>STATE WHICH KIND OF PROOF YOU'RE DOING!!!</u>

3

Reducibility $(A \leq B)$

- Assume B is decidable
- Build machine that uses B to decide A

Mapping Reducibility $(A \leq_m B)$

- Build a machine M' that takes input x and returns y s.t.

$$x \in A \longleftrightarrow y \in B$$

$$\uparrow$$

$$f(x)$$

STATE WHICH KIND OF REDUCE YOU'RE Doing!!

HW5 P6

$T = \{\langle M\rangle \mid M$ is TM that accepts w^R whenever it accepts $w\}$.

WTS: $A_{TM} \leq_R T$ ← mapping reducibility

Build a machine R that computes f s.t.

$\langle x, M\rangle \in A_{TM} \longleftrightarrow f(x) \in T$

$\langle x, M\rangle \notin A_{TM} \longleftrightarrow f(x) \notin T$

If x is accepted by M $\longleftrightarrow$ $f(x, M) \to M'$ s.t. M' accepts w^R whenever it accepts w

If x is not accepted by M $\longleftrightarrow$ $f(x, M) \to M'$ s.t. M' does not always accept w^R whenever it accept w

$R =$ "On input $\langle M, w\rangle$,

1. Build a machine M' that accepts only 01
2. If ($w \in L(M)$) w is accepted by M,
   → Adjust M' also accept 10
   If $w \notin L(M)$
   → M' rejects 10 (don't modify M')
3. Output M'."

HW8 P6

$T = \{\langle M \rangle \mid M$ is TM that accepts ω^R whenever it accepts $\omega\}$

WTS: $A_{TM} \leq_m T$ (mapping reducibility)

Build a machine R that computes f s.t.

$\langle M, w \rangle \in A_{TM} \Leftrightarrow f(x) \in T$

$\langle M, w \rangle \notin A_{TM} \Leftrightarrow f(x) \notin T$

If x is accepted by M $\longleftrightarrow$ $f(\langle M, w \rangle) = M'$ s.t. M' accepts ω^R whenever it accepts ω

If x is not accepted by M $\longleftrightarrow$ $f(\langle M, w \rangle) = M'$ s.t. M' does not always accept ω^R whenever it accepts ω

R = "On input $\langle M, w \rangle$:

1. Build a machine M' that accepts only 01
2. If $(w \in L(M))$ w is accepted by M, $\rightarrow$ Adjust M' also accept 10
   If $w \notin L(M)$ $\rightarrow$ M' rejects 10 (only accepting 01)
3. Output M'."

$EQ_{CFG} = \{ G, H \mid G$ and H are CFGs and $L(G) = L(H) \}$

→ we know ALL_{CFG} is undecidable

$ALL_{CFG} \leq EQ_{CFG}$

$J \in ALL_{CFG} \longleftrightarrow f(J) \rightarrow G, H \in EQ_{CFQ}$

If $L(J) = \Sigma^* \longleftrightarrow L(\overset{J}{\downarrow}G) = L(H)$

If $L(J) \neq \Sigma^* \longleftrightarrow L(\overset{J}{\downarrow}G) \neq L(H)$

Important part ↓

E = " On input ⟨J⟩ ← CFG

1. Let H be a CFG that generates Σ^* $(L(H) = \Sigma^*)$

~~If L(J) = Σ*~~

2. Output ⟨J, H⟩ "

If $L(J) = \Sigma^*$, then $L(J) = L(H) \rightarrow \langle J, H \rangle \in EQ_{CFG}$

If $L(J) \neq \Sigma^*$ then $L(J) \neq L(H) \rightarrow \langle J, H \rangle \notin EQ_{CFG}$