Homework 6

CSC 445-01: Theory of Computation

Matthew Mabrey, Luke Kurlandski

April 20, 2021

4.3

Construct a TM S that will decide $ALL_{DFA} = \{\langle A \rangle \mid A \text{ is a DFA and } L(A) = \Sigma^*\}$. Inside S the machine will check:

1. If all states of the input DFA $\langle A \rangle$ are final (e.g. $Q = F$), ACCEPT!
2. Else, REJECT!

On any input the constructed TM S now correctly:

1. Halts because Q is defined as a finite set of states and $F \subseteq Q$ so our machine will never halt as it doesn't run any input and just checks equivalence of two finite sets.
2. Accepts on $Q = F$ since the DFA will accept all strings (Σ^*).
3. Rejects on $Q \neq F$ because there would be a non-accepting state the DFA could end in which would produce an unaccepted string, meaning it does not produce Σ^*.

Thus the constructed TM S decides ALL_{DFA} because it correctly accepts or rejects and never halts on any input.

4.4

Construct a TM S that will decide $A\varepsilon_{CFG} = \{\langle G \rangle \mid G \text{ is a CFG that generates } \varepsilon\}$. Inside S the machine will:

1. Convert input CFG $\langle G \rangle$ to a CNF
2. If the new CNF grammar's start state has a transition $S_0 - > \epsilon$, then accept
3. Else, reject

4.11

Construct a TM S that will decide $INFINITE_{PDA} = \{\langle M \rangle \mid M \text{ is a PDA and } L(M) \text{ is infinite }\}$. Inside S the machine will:

1. Non-deterministic search towards all final states
2. If there is a loop transition AND every path that has a loop transition that includes pushing a symbol to stack has a counterpart loop that pops these characters, Accept?
3. Else reject

5.1

Given a TM R that we assume decides EQ_{CFG} we can construct a TM S that decides A_{TM}. Inside S the machine will:

1. Construct a CFG G that generates $\overline{\mathrm{L_x}}$ given A_{TM}'s input $\langle M, w\rangle = x$ ($\overline{\mathrm{L_x}} = \Sigma^*$ iff M doesn't accept w)
2. Construct a CFG H that generates Σ^*
3. Feed $\langle G, H\rangle$ as input to the machine R
   - If R accepts, then S rejects
   - If R rejects, then S accepts

We have created a machine that decides A_{TM}, which is undecidable, using the assumed decidability of R. This is a contradiction so R must be undecidable.

5.4

$A = \{0^n 1^n \mid n \geq 0\}$
$B = \{0^n \mid n \geq 0\}$
$f(x) =$ is computed by a TM M that when it reads a 1, it deletes it. M continues until the end of the input string is reached
A is not regular, B is regular
$A \leq_m B$

5.9

1. Construct a TM T that decides $A = \{\langle M\rangle \mid w^R$ is accepted if w is accepted$\}$ inside a TM S used to decide A_{TM}.
2. Inside S, construct a new TM M_1 from input $\langle\langle M\rangle, w\rangle$. Using input w_1:
   - If $w_1 = w^R$, Accept
   - Else if $w_1 = w$, Run M on w_1
   - Else, Reject
3. Feed M_1 into TM T, if it accepts then M must accept w since T only accepts M if both w^R and w are accepted and we can guarantee w^R is accepted because we constructed M_1 to accept it.

Thus we can use TM T to decide A_{TM}, which is undecidable, so T must also be undecidable.

5.22

We let M be the recognizer of A_{TM} and f be the reduction from A to A_{TM}. We describe a recognizer N for A as follows:

1. $N =$ "On input w:
   - Compute $f(w)$
   - Run M on $f(w)$ and output M's outputs"

Clearly, if $w \in A$ then $f(w) \in A_{TM}$ because f is a reduction from A to A_{TM}. Thus, M accepts $f(w)$ whenever $w \in A$. Therefore, N recognizes A.