CMPSC 464: Introduction to the Theory of Computation **(Date: 09/21/2022)**

Recitation #4 Solution

Instructor: Dr. Young Kun Ko *TAs:* Hamed Mahdavi, Levent Toksoz, Neha Sanjay Rathod, Yuzhang Wang

Problem 1

The claim is wrong. Let $L = \Sigma^*$. This is surely a decidable language, however, any language L' is now a subset of L. Then any undecidable language L' (and we know that undecidable languages exist -e.g HALT_{TM}) will satisfy the precondition of the claim (being a subset of L) but will break the conclusion as L' is not a decidable language.

Problem 2

By contradiction assume that there is a decider H for $HALT_{TM}$:

$$H(\langle M, w\rangle) = \begin{cases} \text{accept, if } M \text{ halts on } w \\ \text{reject, if } M \text{ loops on } w \end{cases}$$

From H we can build the following Turing machine D.

$D =$ "On input $\langle M\rangle$

1. Run H on $\langle M, \langle M\rangle\rangle$.
2. If H accepted then D will enter an infinite loop. If H rejected then D accepts."

What happens if we run D on $\langle D\rangle$?

1. D halts on $\langle D\rangle$, but then H rejected $\langle D, \langle D\rangle\rangle$ and hence D looped on $\langle D\rangle$, contradiction!
2. D loops on $\langle D\rangle$, but then H rejected $\langle D, \langle D\rangle\rangle$ and hence D halted on $\langle D\rangle$, contradiction!

Clearly either 1 or 2 has to happen but in both cases we get a contradiction. This implies that D cannot exist, and so H cannot exist either (D was built from H). This means that $HALT_{TM}$ is undecidable.

Problem 3

(a) We want

$$\tilde{\text{D}}(\langle M\rangle) = \begin{cases} \text{accept, if } M \text{ runs on } \langle M\rangle \text{ for an even number of steps} \\ \text{reject, otherwise} \end{cases}$$

To achieve this, $\tilde{\text{D}}$ with input $\langle D\rangle$ runs H with input $\langle M, \langle M\rangle\rangle$ and accepts/rejects as H does.

(b) If $\tilde{\text{Q}}$ is the set of states of $\tilde{\text{D}}$, then ($\tilde{\text{Q}}$,{odd, even}) can be the set of states of D. Initially D behaves exactly like $\tilde{\text{D}}$, except that D keeps track of the parity (even or odd) of its steps with the help of the second component of Q. The start state is (q_0,even). (q_{accept},even) and (q_{reject},odd) are not halting states. There is one additional move going to the halting states (q_{accept},odd) or (q_{reject},even) respectively.
Remark: Actually, it does not matter whether D accepts or rejects an input. Thus, (q_{accept},odd) and (q_{reject},even) could be merged into one halting state.

(c) Note that D with input $\langle M\rangle$ runs for an odd number of steps, if M on input $\langle M\rangle$ runs for an even number of steps. Otherwise(i.e., if M on input $\langle M\rangle$ runs for an odd number of steps or does not halt), D with input $\langle M\rangle$ runs for an even number of steps.

Now, run the TM D on the input $\langle D \rangle$. D on $\langle D \rangle$ runs for an even number of steps iff D on $\langle D \rangle$ does not run for an even number of steps.
This is a contradiction. Thus, the assumption that H exists is false. (Thus also $\tilde{\mathrm{D}}$ and D don't exist.)
(Because, D halts on all inputs, we also know that D on $\langle D \rangle$ runs for an even number of steps iff D on $\langle D \rangle$ runs for an odd number of steps.)