CPSC 421: Introduction to Theory of Computing **Winter Term 1 2018-19**

Lecture 17: October 17

Lecturer: Nicholas Harvey *Scribes: Kaitian Xie*

17.1 Reductions

Recall:

- $A_{TM} = \{\langle M, w\rangle : M \text{ accepts } w\}$
- $HALT_{TM} = \{M \text{ halts on } w\}$

Theorem 17.1 *$HALT_{TM}$ is undecidable.*

Proof: Last time showed that A_{TM} is undecidable. Show that $A_{TM} \leq_T HALT_{TM}$. Suppose we have a TM R that decides $HALT_{TM}$. Then we want to create a TM S that decides A_{TM}.

Design S: On input x,

1. Reject if x not in form of $\langle M, w\rangle$.
2. Run R on input $\langle M, w\rangle$.
3. If R accepts, (we know M halts on input w), simulate M on input w. Accept if M accepts, reject if M rejects. Else (M runs forever on input w). Reject. (M does not accept w)

So S is a decider for A_{TM}. Contradiction! ■

Claim 17.2 *Suppose L and $\overline{L}$ are both recognizable, then L is decidable (and so is $\overline{L}$.*

Proof: Let M_1 be TM that recognizes L. Let M_2 be TM that recognizes $\overline{L}$. Design a new TM M_3 as follows: on input x,

1. In parallel simulate both M_1 and M_2.
2. If M_1 halts and accepts then M_3 accepts.
3. If M_2 halts and accepts then M_3 rejects.

Suppose $x \in L$. Then M_1 eventually accepts $x \Rightarrow M_3$ accepts x. $x \notin L$, then M_2 eventually accepts $x \Rightarrow M_3$ rejects x. So M_3 decides L. ■

Corollary 17.3 *$\overline{A_{TM}}$ is not recognizable.*

Proof: If it was recognizable, Claim should imply A_{TM} is decidable! ■