Homework 6

March 3, 2017

(804501476)

Problem 1.0:
Prove

$$COMP_{TM} = \{< M_1, M_2 > \mid L(M_1) = \overline{L(M_2)}\} \tag{1.1}$$

Proof. Towards contradiciton assume $COMP_{TM}$ is recognizable

- $\exists$ turing machine R which recognizes $COMP_{TM}$.
- Build turing machine N
- $N(y)$:
  - Let $x =< N >$ // By Recursion Theorem
  - Let $< P >$ be a turing machine that recognizes prime numbers.
  - If $y = 4$ Accept.
  - Run $R(x, < P >)$
    1. $R(x, < P >) : Accepts \quad //R$ thinks $L(N) = L(\bar{P})$
       - Run $P(y) \implies L(N) = L(P) \Longrightarrow\!\Longleftarrow$
    2. $R(x, < P >) : Rejects \quad //R$ thinks $L(N) \neq L(\bar{P})$
       - Run $\bar{P}(y) \implies L(N) = L(\bar{P}) \Longrightarrow\!\Longleftarrow$
    3. $R(x, < P >) : Loops \quad //R$ thinks $L(N) \neq L(\bar{P})$
       - But by construction $L(N) = 4 \in L(\bar{P}) \Longrightarrow\!\Longleftarrow$

□

Problem 2.0:
Prove

$$NEQ_{TM} = \{< M_1, M_2 > \mid L(M_1) \neq L(M_2)\} \tag{2.2}$$

Proof. Towards contradiciton assume NEQ_{TM} is recognizable

- $\exists$ turing machine R which recognizes NEQ_{TM}.
- Build turing machine N
- $N(y)$:
  - Let $x =< N >$ // By Recursion Theorem
  - Let $< M >$ be a turing machine that rejects everything.
  - If $y = \varepsilon$, accept.
  - Run $R(x, < M >)$
    1. $R(x, < M >) : Accepts \quad //R$ thinks $L(N) \neq L(M)$
       - Reject all $y \implies L(N) = 0 \Longrightarrow\Longleftarrow$
    2. $R(x, < M >) : Rejects \quad //R$ thinks $L(N) = L(M)$
       - Accept all $y \implies L(N) \neq 0 \Longrightarrow\Longleftarrow$
    3. $R(x, < M >) : Loops \quad //R$ thinks $L(N) = L(N)$
       - But by construction $L(N) = \{\varepsilon\} \implies L(N) \neq 0 \Longrightarrow\Longleftarrow$

□

Problem 3.0:
A certified language is a language over $\{0, 1\}$ s.t there exists a turing machine M satisfying the following conditions:

- $\forall\ x \in L,\ \exists\ y \in \{0,1\}^*\ s.t\ M(x, y) : accepts$
- $\forall\ x \notin L,\ and\ \forall\ y \in \{0,1\}^*\ s.t\ M(x, y) : Rejects$

a: Show $Halt_\varepsilon$ is a certified language

Let $L' = \{w \mid w$ is the number of steps in TM which halts on $\varepsilon\}$ and let $y \in L'$.
We construct $M(x, y)$:

- Run $x(\varepsilon)$ one step at at time.
  - If $x(\varepsilon)$ is in accept state after y steps: **accept**
  - If the number of steps in $x(\varepsilon)$ is greater than y: **reject**

The machine satisfies property 1) if $x \in L$ then $\exists\ y \in L'$ which contains the number of steps to compute $x(\varepsilon)$.
The machine also satisfies property 2) since if $x \notin L$ then there is no solution to $x(\varepsilon)$ since x does not halt. Therefore $M(x, y)$ will reject for all y.

b)

The number of inputs to machine x is infinite and each one of these inputs produces a different number of steps for a computationl path. Therfore the number of steps is also infinite so we would never be able to verify if $x \in Halt_{all}$

c)

Proof. Assume for contradiction that $Halt_{all}$ is certifiable then ∃ M that certifies it.
Construct N(x):

- Let $z =< N >$ / / by Recursion Theorem
- Run M(z,x)

  1 M:accepts $\implies$ M certifies N, so N halts.

  - Loop $\implies\impliedby$

  2 M:rejects $\implies$ M does not certify N, so N loops.

  - Halt $\implies\impliedby$

Therefore $Halt_{all}$ is not certifiable. □