CMPS 130: HW 3

Kevin Wang

March 2, 2019

Solution to Problem 1

Definition 1.1 (Kleene Closure). Σ^* is the set of all finite strings over Σ as defined by $\bigcup_{k \geq 0} \Sigma^k$.[1]

Definition 1.2 (Recursively Enumerable Languages). A language $\mathcal{L}$ is Recursively Enumerable (RE) if there exists some Turing Machine M, such that $L(M) = \mathcal{L}$.[2]

Theorem 1. *Recursively Enumerable (RE) languages are closed under Kleene Closure.*

Proof. Let $\mathcal{L}$ be some RE language and let M be some Turing Machine (TM) such that $L(M) = \mathcal{L}$ (Definition 1.2). Let $\mathcal{L}^*$ be the Kleene Closure of $\mathcal{L}$.

We define a TM M^* that receives input x. M^* then non-deterministically splits x into $s_1, s_2, \cdots, s_k$ where $k \leq |x|$. Note that the number of ways to split x is finite due to the length of input x. Input x is then accepted if for all $i \leq k$, s_i is accepted. The trivial case, $x = \epsilon$, is also accepted.

Observe that M^* simulates the Kleene Closure (Definition 1.1) and therefore accepts $\mathcal{L}^*$, such that $L(M^*) = \mathcal{L}^*$. Thus, $\mathcal{L}^*$ is Recursively Enumerable (Definition 1.2) – proving that RE languages are closed under Kleene Closure. $\square$

---

[1] lec2.pdf
[2] lec12.pdf

Solution to Problem 2

Theorem 2.

Proof. $\square$

Solution to Problem 3

Theorem 3.

Proof. ☐

Solution to Problem 4

Theorem 4.

Proof. ∎

Solution to Problem 5

Theorem 5.

Proof. □

Solution to Problem 6

Theorem 6.

Proof. □