Computing Functions

Definition 5.2. Suppose that Σ and Γ are alphabets that do not contain **#** and that f is a function from Σ^* to Γ^*. We say that f is ***Turing-computable*** if there is a Turing machine $M = (Q, \Lambda, q_0, \delta)$ such that $\Sigma \subseteq \Lambda$ and $\Gamma \subseteq \Lambda$ and for each string $w \in \Sigma^*$, when M is run with input w, it halts with output $f(w)$. In this case, we say that M ***computes*** the function f.

Computable Languages

Definition 5.3. Let Σ be an alphabet that does not contain **#** and let L be a language over Σ. We say that L is ***Turing-decidable*** if there is a Turing machine $M = (Q, \Lambda, q_0, \delta)$ such that $\Sigma \subseteq \Lambda$, $\{0, 1\} \subseteq \Lambda$, and for each $w \in \Sigma^*$, when M is run with input w, it halts with output $\chi_L(w)$. (That is, it halts with output 0 or 1, and the output is 0 if $w \notin L$ and is 1 if $w \in L$.) In this case, we say that M ***decides*** the language L.

Definition 5.4. Let Σ be an alphabet that does not contain **#**, and let L be a language over Σ. We say that L is ***Turing-acceptable*** if there is a Turing machine $M = (Q, \Lambda, q_0, \delta)$ such that $\Sigma \subseteq \Lambda$, and for each $w \in \Sigma^*$, M halts on input w if and only if $w \in L$. In this case, we say that M ***accepts*** the language L.

6. Draw a transition diagram for a Turing machine which decides the language $\{a^n b^n \mid n \in \mathbb{N}\}$. Explain in general terms how to make a Turing machine that decides the language $\{a^n b^n c^n \mid n \in \mathbb{N}\}$.

7. Draw a transition diagram for a Turing machine which decides the language $\{a^n b^m \mid n > 0 \text{ and } m \text{ is a multiple of } n\}$.

8. Based on your answer to the previous problem and the copying machine presented in this section, describe in general terms how you would build a Turing machine to decide the language $\{a^p \mid p \text{ is a prime number}\}$.