CS 103: Mathematical Foundations of Computing
Problem Set #8

Quang Nguyen

June 1, 2022

Due Friday, November 19 at 2:30 pm Pacific

Do not put your answers to Problems 1, 2.ii, 3.ii, or 4 in this file.
Symbols you might find helpful in this problem set:

- Language derivatives use the notation $\partial_w L$.
- Distinguishability is written as $x \not\equiv_L y$.

Problem Two: The Complexity of Addition

i.

Claim: The language ADD defined above is regular.
Disproof: We will show that the negation of this statement is true, namely, that the language ADD defined above is not regular. Let $S = \{1^x | x \in \mathbb{N}\}$. We will prove that S is infinite and that S is a distinguishing set for ADD.
To see that S is infinite, note that S contains one string for each natural number.
To see that S is a distinguishing set for ADD, consider any string $1^x, 1^y \in S$ where $x \neq y$. Note that $1^x + 1^y = 1^{x+y} \in ADD$ and that $1^y + 1^y = 1^{x+y} \notin ADD$. Thus $1^m \not\equiv_{ADD} 1^n$. Therefore, by the Myhill-Nerode theorem, we see that ADD is not regular. ■

Problem Three: Brzozowski's Theorem

i.

1. aab
2. bbbaaa
3. aab

iii.

Theorem: Let L be a language over Σ and let $S \subseteq \Sigma^\star$ be a set containing infinitely many strings. If the following statement is true, then L is not regular:

$$\forall x \in S. \forall y \in S. (x \neq y \rightarrow \partial_x L \neq \partial_y L)$$

Proof: Assume that the above statement is true. Notice, by definition, that S contains infinitely many strings. We will prove that S is a distinguishing set for L.
Pick arbitrary $x \in S$ and $y \in S$ such that $x \neq y$. We see that $\partial_x L \neq \partial_y L$. This means that there is an element w in $\partial_x L$ and not in $\partial_y L$, or vice versa. Assume, without loss of generality, that $w \in \partial_x L$ and $w \notin \partial_y L$. This means that $xw \in L$ and $yw \notin L$. Thus $x \not\equiv_L y$. Therefore, by Myhil-Nerode theorem, we see that L is not regular. ■

Problem Five: Executable Computability Theory

i.

- $L = \{a^n b^n | n \in \mathbb{N}\}$
- The function is a decider for the language.
- L is decidable.

ii.

- $L = \{w \in \{a, b\}^\star\}$
- The function is not a decider for the language.
- L is decidable.

iii.

- $L = \{w \in \{a, b\}^\star || w| \text{ is a Fibonacci number}\}$
- The function is not a decider for the language.
- L is decidable.

Problem Six: What Does it Mean to Solve a Problem?

i.

Let L be any language over Σ and M be a machine that always rejects. This TM satisfies property (1) because it always halts. It also vacuously satisfies property (2) because the machine never accepts anything.

ii.

Let L be any language over Σ and M be a machine that always accepts. This TM satisfies property (1) because it always halts. It also vacuously satisfies property (2) because the machine never rejects anything.

iii.

Let L be any language over Σ and M be a machine that always loops. This TM vacuously satisfies properties (2) and (3) because it never halts, meaning that it never accepts or rejects anything.

iv.

Theorem: If there is a TM for L meeting the three above criteria, then $L \in R$.
Disproof: Consider any machine M matching the above criteria. First, by property (1), we know that M halts on all inputs. Next, we know, by properties (2) and (3), that M accepts a string w if and only if $w \in L$. Therefore, L is decidable, as required. ■