Homework 1: Strings and Languages

Will Griffin

January 21, 2026

1. Proof Practice

a. Statement-Reason Proof:

To show: If s is a string, every substring of a substring of s is a substring of s.

Proof:

1. y is a substring of s	Universal instantiation
2. $s = xyz$ for some x, z	(1), def. substring
3. v is a substring of y	Universal instantiation
4. $y = uvw$ for some u, w	(3), def. substring
5. $s = xuvwz$	(2), (4), substitution

b. Paragraph Proof:

To show: If w is a string, every prefix of a suffix of w is a suffix of a prefix of w.

Proof: Let v be a suffix of w, that is, $w = xv$ for some x; and let y be a prefix of v, that is, $v = yz$ for some z. We can combine these equations such that $w = xyz$. By definition of a prefix, xy is a prefix of w, and by definition of a suffix, y is a suffix of xy. Therefore, y is a suffix of a prefix xy of w.

2. String homomorphisms

To show: For any string homomorphism f, and for any string $w = w_1 \cdots w_n$ (where $n \geq 0$ and, for $j = 1, \ldots, n, w_j \in \Sigma$), we have

$$f(w) = f(w_1) \cdots f(w_n). \tag{$*$}$$

You may assume the following about strings:

- Identity: For all $x \in \Sigma^*$, $x\varepsilon = x$ and $\varepsilon x = x$.
- Right cancellation: For all x, y, $z \in \Sigma^*$, if $xz = yz$, then $x = y$.
- Left cancellation: For all x, y, $z \in \Sigma^*$, if $xy = xz$, then $y = z$.

Proof: (a) Let $n = 0$. Any string of length 0 is ε. Therefore, $w = \varepsilon$. for an arbitrary string u concatenated with w, that is, $uw = u\varepsilon$, any string concatenated with the empty string is itself by the given identity axiom. Therefore, $uw = u\varepsilon = u$.

For a homomorphism of uw, that is, $f(uw)$, since $uw = u$, $f(uw) = f(u)$. Therefore, since $w = \varepsilon$, $f(w) = f(\varepsilon) = \varepsilon$. Therefore, $(*)$ is true for $n = 0$.

(b) Assume $(*)$ is true for $n = i$, where $i \geq 0$. Therefore, $w = w_1 \cdots w_i$ and $f(w) = f(w_1 \cdots w_i) = f(w_1) \cdots f(w_i)$.

Let $n = i + 1$. Therefore, $w = w_1 \cdots w_i w_{i+1}$. This can be written as a concatenation of a string of length i and a string of length 1, that is, $w = (w_1 \cdots w_i)w_{i+1}$.

Applying the homomorphism function, we get $f(w) = f((w_1 \cdots w_i)w_{i+1})$. This can be expanded to $f(w_1 \cdots w_i)f(w_{i+1})$ by our definition of a string homomorphism.

Applying the inductive hypothesis, we get $f(w) = (f(w_1) \cdots f(w_i))\, f(w_{i+1})$. Removing the parentheses, this becomes $f(w) = f(w_1) \cdots f(w_i)f(w_{i+1})$. This falls in the original form $f(w) = f(w_1) \cdots f(w_n)$.

Since a string homomorphism f is shown to operate symbol-by-symbol for a string of length 0 and a string of length $n = i + 1$ when it is assumed true for $n = i$, by induction, this is true of any string with length $n \geq 0$.

3. Finite and cofinite

Let $\Sigma = \{a, b\}$. Define FINITE to be the set of all finite languages over Σ, and let $\mathsf{coFINITE}$ be the set of languages over Σ whose *complement* is finite:

$$\mathsf{coFINITE} = \{L \subseteq \Sigma^* | \overline{L} \in \mathsf{FINITE}\}$$

where $\overline{L} = \Sigma^* \setminus L$. For example, Σ^* is in $\mathsf{coFINITE}$ because its complement is $\emptyset$, which is finite. (Please think carefully about this definition, and note that $\mathsf{coFINITE}$ isn't the same thing as $\overline{\mathsf{FINITE}}$).

You may assume the union of two finite sets is finite.

(a) If $L \in \mathsf{FINITE}$, what data structure could you use to represent L, and given a string w, how would you decide whether $w \in L$?

To represent L, I would use a trie data structure (a tree used to store letters). To decide whether $w \in L$, starting at the root, for each character $c \in w$, search for c in the current node's children, and if found, set that child as the current node. If all characters of w are found in order, $w \in L$. Otherwise, if a character is not found, $w \notin L$.

(b) If $L \in \mathsf{coFINITE}$, what data structure could you use to represent L, and given a string w, how would you decide whether $w \in L$?

I would use a trie to store $\overline{L}$. Since $L \in \mathsf{coFINITE}$, its complement would have a finite number of elements and therefore would be able to be stored in a finite data structure in some kind of memory. To decide whether $w \in L$, starting at the root, for each character $c \in w$, search for c in the current node's children, and if found, set that child as the current node. If all characters of w are found in order, $w \in \overline{L}$, and therefore $w \notin L$. Otherwise, if a character is not found, $w \in L$.

(c) Are there any languages in $\mathsf{FINITE} \cap \mathsf{coFINITE}$? Prove your answer.

L is a finite language, that is, $L \in \mathsf{FINITE}$. Since Σ^* is infinite and L is finite, $\Sigma^* \setminus L$ is by definition infinite. Therefore, since $\overline{L} = \Sigma^* \setminus L$, $\overline{L}$ is infinite. Since $\overline{L}$ is infinite, $L \notin \mathsf{coFINITE}$. Therefore, $\mathsf{FINITE} \cup \mathsf{coFINITE} = \emptyset$.

(d) Are there any languages over Σ that are *not* in $\mathsf{FINITE} \cup \mathsf{coFINITE}$? Prove your answer.

For the sake of proof by counterexample, let language L be the set of all strings of only "a", that is, $L = \{a, aa, aaa, aaaa, ...\}$. Therefore, $\overline{L} = \{\varepsilon, b, ab, ba, bb, ...\}$.

Since L is infinite, $L \notin \mathsf{FINITE}$. Since $\overline{L}$ is infinite, $L \notin \mathsf{coFINITE}$. Therefore, $L \notin \mathsf{FINITE} \cup \mathsf{coFINITE}$.

Therefore, there are langauges that are not in $\mathsf{FINITE} \cup \mathsf{coFINITE}$.