Assignment 1: Languages, Induction, Recursion

Return by 11:59pm 9/12/2019

Problem 1 5%

The language *Balanced* over $\Sigma = \{(,)\}$ is defined recursively as follows

1. $\Lambda \in$ *Balanced*.
2. $\forall x, y \in$ *Balanced*, both xy and (x) are elements of *Balanced*.

A prefix of a string x is a substring of x that occurs at the beginning of x. Prove by induction that a string x belongs to this language if and only if (iff) the statement $B(x)$ is true.
$B(x)$: x contains equal numbers of left and right parentheses, and no prefix of x contains more right than left.
Reminder for this and all following assignments: if you need to prove the "iff" statement, i.e., $X \iff Y$, you need to prove both directions, namely, "given X, prove that Y follows from X ($X \implies Y$)", and "given Y, prove that X follows from Y ($X \impliedby Y$)".

Problem 2 5%

Complete proof of claim about the reverse function (see Lecture 3).

Problem 3 5%

Finite language is a language with finite number of strings in it, i.e., there exist exactly k strings in this language such that $k \in \mathbb{N}$ and $k \neq \infty$. For a finite language L, let $|L|$ denote the number of elements of L. For example, $|\{\Lambda, a, ababb\}| = 3$. (Do not mix up with the length $|x|$ of a string x.) The statement $|L_1L_2| = |L_1||L_2|$ says that the number of strings in the concatenation L_1L_2 is the same as the product of the two numbers $|L_1|$ and $|L_2|$. Is this always true? If so, prove, and if not, find two finite languages $L_1, L_2 \subseteq \{a, b\}^*$ such that $|L_1L_2| \neq |L_1||L_2|$.

Problem 4 5%

We proved in class that if L_1, and L_2 are subsets of $\{a,b\}^*$ then $L_1^* \cup L_2^* \subseteq (L_1 \cup L_2)^*$. Show that $L_1^* \cup L_2^* \neq (L_1 \cup L_2)^*$.

Problem 5 5%

Find an example of languages L_1 and L_2 for which neither of L_1, L_2 is a subset of the other, but $L_1^* \cup L_2^* = (L_1 \cup L_2)^*$. Prove the correctness of your example.

Problem 6 10%

Given language $L = \{yy | y \in \{a,b\}^*\}$. L can be represented as a concatenation

$$L = L\{\Lambda\} = \{\Lambda\}L$$

like any language. Can you express L as $L = L_1 L_2$, where $L_1 \neq \{\Lambda\}$, and $L_2 \neq \{\Lambda\}$? Prove your answer.

Problem 7 5%

Each case below gives a recursive definition of $L \subseteq \{a,b\}^*$. Give a simple nonrecursive definiton of L in each case. Example: $a \in L;\ \ \forall x \in L\ \ \ ax \in L$ can be defined as "The set of all non-empty strings that do not contain b."

1. $a \in L;\ \ \forall x \in L\ \ \ xa, xb \in L$
2. $a \in L;\ \ \forall x \in L\ \ \ bx, xb \in L$
3. $a \in L;\ \ \forall x \in L\ \ \ ax, xb \in L$
4. $a \in L;\ \ \forall x \in L\ \ \ xb, xa, bx \in L$
5. $a \in L;\ \ \forall x \in L\ \ \ xb, ax, bx \in L$
6. $a \in L;\ \ \forall x \in L\ \ \ xb, xba \in L$

Problem 8 5%

Suppose that Σ is an alphabet, and that $f : \Sigma^* \to \Sigma^*$ has the property that $f(\sigma) = \sigma$ for every $\sigma \in \Sigma$ and $f(xy) = f(x)f(y)$ for every $x, y \in \Sigma^*$. Prove that for every $x \in \Sigma^*$, $f(x) = x$.

Problem 9 15%

In each case below, find a recursive definition for the language L and prove that it is correct.

1. $L = \{a^i b^j | j \geq 2i\}$
2. $L = \{a^i b^j | j \leq 2i\}$

Problem 10 10%

Suppose $L \subseteq \{a, b\}^*$ is defined as follows: $\Lambda \in L$; for every x and y in L, the strings axb, bxa, and xy are in L. Show that $L = AEqB$, the language of all strings x in $\{a, b\}^*$ satisfying $n_a(x) = n_b(x)$.

Problem 11 10%

Let L_1, L_2, and L_3 be languages over some alphabet. In each case below, two languages are given. Say what the relationship is between them. (Are they always equal? If not, is one always a subset of the other?) Give reasons for your answers, including counterexamples if appropriate.

1. $L_1(L_2 \cap L_3)$, $L_1L_2 \cap L_1L_3$
2. $L_1^* \cap L_2^*$, $(L_1 \cap L_2)^*$
3. $L_1^* L_2^*$, $(L_1L_2)^*$

Problem 12 10%

For $x \in EXPR$ defined in class, $n_a(x)$ denotes the number of a's in the string, and we will use $n_{op}(x)$ to stand for the number of operators in x (the number of occurrences of $+$ or $*$). Show that for every $x \in EXPR$, $n_a(x) = 1 + n_{op}(x)$.

Problem 13 10%

Show using induction that for every $x \in \{a, b\}^*$ such that x begins with a and ends with b, x contains the substring ab.