CS 520
Fall 2017
Homework #1

Due in class Friday, September 22, 2017

Rules for Homework.

i. You may discuss these problems with others, but what you write must be yours and yours alone. Use of any sources other than class notes and recommended texts must be accompanied by a citation.

ii. Please justify any assertions you make.

ii. Remember that you will be graded on correctness *and* clarity.

1 Let X be a set. A binary relation $\sim$ is called an *equivalence relation* if:

i) $x \sim x$;

ii) $x \sim y$ implies $y \sim x$;

iii) $x \sim y$ and $y \sim z$ implies $x \sim z$.

A function f with domain X is called a *representing function* for $\sim$ if $x \sim y$ exactly when $f(x) = f(y)$.

a) Let $n > 1$ be an integer. Find a representing function when $x \sim y$ iff $x - y$ is a multiple of n.

b) Show that every equivalence relation has a representing function.

c) Suppose X has n elements. Consider two methods for storing an equivalence relation on a computer: a) a 0-1 array indicating, for each pair, whether the relation holds between them or not; b) a succinct tabulation of the representing function. Which uses less space?

2 Let a, b be constants. The n-th Gibbonacci number is defined by the relations

$$\begin{aligned} G_0 &= a; \\ G_1 &= b; \\ G_n &= G_{n-1} + G_{n-2}, \text{ for } n \geq 2. \end{aligned}$$

Find a formula for the determinant

$$\begin{vmatrix} G_{n+2} & G_{n+1} \\ G_{n+1} & G_n \end{vmatrix}.$$

You should use mathematical induction to check it (but you need not turn that in).

3 Algebraic expressions are sometimes represented using *postfix notation*, in which the operator is written after its operands. This can be defined formally, but the following examples should serve to illustrate the idea:

standard form	postfix
$\texttt{x} + \texttt{y}$	$\texttt{xy}+$
$(\texttt{a} + \texttt{b}) * \texttt{c}$	$\texttt{ab} + \texttt{c}*$
$(\texttt{a} + \texttt{b}) - (-\texttt{c})$	$\texttt{ab} + \texttt{c} - -$

Note that in postfix notation, no parentheses are needed.

To avoid ambiguity, parentheses have been used in our definition of regular expressions. But this could also have been done using postfix notation.

a) Consider the following (unabbreviated) regular expressions, which were examples in the text:

$$\texttt{0} * \texttt{10}*, \qquad \texttt{1}(\texttt{01}+)*, \qquad \texttt{01u10}.$$

(Here `u` is used as a "boldface union" symbol.) Restore the parentheses in each, and then convert it into postfix form.

b) Give a recursive definition of the postfix form of a regular expression.

c) Using your definition, prove by induction that the length of a postfix regular expression is no longer than the length of the standard (parenthesized) regular expression you started with.

4 Here is Kleene's original example of a non-regular language. The symbols are $\{0, 1\}$. A string $x_1 x_2 \cdots x_n$ belongs to L if (and only if) $x_i = 1$ exactly when i is a perfect square. For example,

$$1001 \in L$$

but

$$1111 \notin L.$$

Prove that L is not regular.