Extra Practice Problems 9

Here's yet another batch of practice problems. If you'd like even more practice, please let us know what topics you'd like more review on!

Problem One: Set Theory

Prove or disprove: there are sets A and B where $\wp(A \times B) = \wp(A) \times \wp(B)$.

Problem Two: Induction

The ***well-ordering principle*** states that if $S \subseteq \mathbb{N}$ and $S \neq \emptyset$, then S contains an element n_0 that is less than all other elements of S. There is a close connection between the well-ordering principle and the principle of mathematical induction.

Suppose that P is some property such that

- $P(0)$
- $\forall k \in \mathbb{N}. (P(k) \rightarrow P(k+1))$

Using the well-ordering principle, *but without using induction*, prove that $P(n)$ holds for all $n \in \mathbb{N}$. This shows that if you believe the well-ordering principle is true, then you must also believe the principle of mathematical induction.

Problem Three: Graphs

Let $G = (V_1, E_1)$ and $H = (V_2, E_2)$ be undirected graphs. The ***tensor product*** of G and H, denoted $G \times H$, is an undirected graph. $G \times H$ has as its set of nodes the set $V_1 \times V_2$. The edges of $G \times H$ are defined as follows: the edge $\{(u_1, v_1), (u_2, v_2)\}$ is in $G \times H$ if $\{u_1, u_2\} \in E_1$ and $\{v_1, v_2\} \in E_2$.

Prove that $\chi(G \times H) \leq \min\{\chi(G), \chi(H)\}$.

Interestingly, the following question is an open problem: are there any undirected graphs G and H for which $\chi(G \times H) \neq \min\{\chi(G), \chi(H)\}$? A conjecture called *Hedetniemi's conjecture* claims that the answer is no, but no one knows for sure!

Problem Four: First-Order Logic

Consider the following formula in first-order logic:

$$\forall x \in \mathbb{R}.\ \forall y \in \mathbb{R}.\ (x < y \to \exists p \in \mathbb{Z}.\ \exists q \in \mathbb{Z}.\ (q \neq 0 \land x < p/q \land p/q < y))$$

This question explores this formula.

i. Translate this formula into plain English. As a hint, there's a very simple way of expressing the concept described above.

ii. Rewrite this formula so that it doesn't use any universal quantifiers.

iii. Rewrite this formula so that it doesn't use any existential quantifiers.

iv. Rewrite this formula so that it doesn't use any implications.

v. Negate this formula and push the negations as deep as possible.

Problem Five: Binary Relations

Officially, a binary relation R over a set A is just a subset of A^2 consisting of the ordered pairs (a, b) such that aRb. For example, the binary relation $<$ over the set $\{0, 1, 2, 3\}$ is the set

$$\{(0, 1), (0, 2), (0, 3), (1, 2), (1, 3), (2, 3)\}.$$

Because we can think of relations as sets, we can apply set-theoretic operations to them.

i. Prove or disprove: if R_1 and R_2 are equivalence relations over a set A, then $R_1 \cap R_2$ is an equivalence relation over A.

ii. Prove or disprove: if R_1 and R_2 are equivalence relations over a set A, then $R_1 \cup R_2$ is an equivalence relation over A.

iii. Prove or disprove: if R_1 and R_2 are strict orders over a set A, then $R_1 \cap R_2$ is a strict order over A.

iv. Prove or disprove: if R_1 and R_2 are strict orders over a set A, then $R_1 \cup R_2$ is a strict order over A.

Problem Six: Functions and Relations

In this question, let $A = \{1, 2, 3, 4, 5\}$.

Let $f : A \to A$ be an arbitrary function from A to A that we know is ***not a surjection***. We can then define a new binary relation $\sim_f$ as follows: for any $a, b \in A$, we say $a \sim_f b$ if $f(a) = b$. Notice that this relation depends on the particular non-surjective function f that we pick; if we choose f differently, we'll get back different relations. This question explores what we can say with certainty about $\sim_f$ knowing only that its domain and codomain are A and that it is not a surjection.

Below are the six types of relations we explored over the course of this quarter. For each of the types, determine which of the following is true:

- The relation $\sim_f$ is ***always*** a relation of the given type, regardless of which non-surjective function $f : A \to A$ we pick.
- The relation $\sim_f$ is ***never*** a relation of the given type, regardless of which non-surjective function $f : A \to A$ we pick.
- The relation $\sim_f$ is ***sometimes, but not always*** a relation of the given type, depending on which particular non- surjective function $f : A \to A$ we pick.

Since these options are mutually exclusive, check only one box per row. *(Hint: Draw a lot of pictures.)*

$\sim_f$ is reflexive	☐ ***Always***	☐ ***Sometimes, but not always***	☐ ***Never***
$\sim_f$ is irreflexive	☐ ***Always***	☐ ***Sometimes, but not always***	☐ ***Never***
$\sim_f$ is symmetric	☐ ***Always***	☐ ***Sometimes, but not always***	☐ ***Never***
$\sim_f$ is asymmetric	☐ ***Always***	☐ ***Sometimes, but not always***	☐ ***Never***
$\sim_f$ is transitive	☐ ***Always***	☐ ***Sometimes, but not always***	☐ ***Never***
$\sim_f$ is an equivalence relation	☐ ***Always***	☐ ***Sometimes, but not always***	☐ ***Never***
$\sim_f$ is a strict order	☐ ***Always***	☐ ***Sometimes, but not always***	☐ ***Never***

Problem Seven: The Pigeonhole Principle*

Let n be an odd natural number and consider the set $S = \{1, 2, 3, \ldots, n\}$. A ***permutation*** of S is a bijection $\sigma : S \to S$. In other words, σ maps each element of S to some unique element of S and does so in a way such that no two elements of S map to the same element.

Let σ be an arbitrary permutation of S. Prove that there is some $r \in S$ such that $r - \sigma(r)$ is even.

Problem Eight: DFAs, NFAs, and Regular Expressions

If w is a string, then w^R represents the reversal of that string. For example, the reversal of "table" is "elbat." If L is a language, then L^R is the language $\{ w^R \mid w \in L \}$ consisting of all the reversals of the strings in L.

It turns out that the regular languages are closed under reversal.

i. Give a construction that turns an NFA for a language L into an NFA for the language L^R. No proof is necessary.

ii. Give a construction that turns a regular expression for a language L into a regular expression for the language L^R. No proof is necessary.

Problem Nine: Nonregular Languages

Prove that the language $\{ w \in \{\mathtt{a}, \mathtt{b}\}^* \mid |w| \equiv_3 0$ and the middle third of the characters in w contains at least one $\mathtt{a}$ $\}$ is not regular.

Problem Ten: Context-Free Grammars

Let $\Sigma = \{\mathtt{(}, \mathtt{)}\}$ and let $L = \{ w \in \Sigma^* \mid w$ is a string of balanced parentheses and w has an even number of open parentheses $\}$. Write a CFG for L.

Problem Eleven: Turing Machines

Design a TM subroutine with the following behavior: if the TM is initialized so that the tape stores a base-10 number n with the tape head at the first digit of that n, the TM ends with the number $n+1$ written on the tape, with the tape head positioned at the first digit of $n+1$.

* Adapted from http://www.cut-the-knot.org/do_you_know/pigeon.shtml.

Problem Twelve: R and RE Languages

Given any computable function f and language L, let's define $f[L] = \{ w \in \Sigma^* \mid \exists x \in L.\ f(x) = w \}$. In other words, $f[L]$ is the set of strings formed by applying f to each string in L.

Prove that if $L \in$ **RE** and f is a computable function, then $f[L] \in$ **RE**.

Problem Thirteen: Impossible Problems

Prove that $L = \{ \langle M, N \rangle \mid M \text{ is a TM, } N \text{ is a TM, and } \mathscr{L}(M) = \overline{\mathscr{L}(N)} \}$ is not in **RE**.

Problem Fourteen: P and NP

Suppose that V is a polynomial-time verifier for an **NP**-complete language L. That is,

$$w \in L \quad \text{iff} \quad \exists c \in \Sigma^*.\ V \text{ accepts } \langle w, c \rangle$$

and

$$V \text{ runs in time polynomial in } |w|$$

Now, suppose that V is a "superverifier" with the property that for any string $w \in L$, V accepts $\langle w, c \rangle$ for almost all choices of c. Specifically, for any $w \in L$, there are at most five strings c for which V rejects $\langle w, c \rangle$.

Under these assumptions, prove that **P** = **NP**.