CS/Math 240 Fall 2012
Homework 4

Due: Monday November 19th, 9:55am room 4382 CS (or, discussion/review at that time).

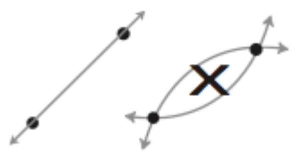
Problem 1. Hilbert's Axioms for Geometry

The Predicate logic gives us a way to differentiate between saying that something is true *for all* and that it is true *for some*. This is done through the use of *predicates* and *quantifiers*.

The mathematician David Hilbert used Predicate logic to formalize a set of axioms for geometry. Here are 3 of his axioms.

Axioms of Incidence

1. For every two points A and B, there exists a *unique* line ℓ that contains both of them.



2. There are at least two points on any line.

3. There exist at least three points that do not all lie on a line.

To state these axioms in the Predicate logic we need a predicate C(x,A) which is true exactly when the point A lies on the line x. We can use this predicate together with the quantifiers *there exists* (∃) and *for all* (∀) to formally state Hilbert's axioms. Each time we use a quantifier we also need to identify the set of objects that are its universe. We will use Lines to name the set of all lines and Points to name the set of all points. With this, Hilbert's second axiom can be written as follows.

(∀x ∈ Lines) [(∃A ∈ Points) (∃ B ∈ Points) [A ! = B & C(x,A) & C(x,B)]

Write axioms 1 and 3 using Predicate logic.

Problem 2.

Suppose there is a circular auto track that is one mile long. Along the track there are n > 0 gas stations. The combined amount of gas in all gas stations allows a car to travel exactly one mile. A car has a gas tank that will hold more than a gallon of gas, but it starts out empty. Show that no matter how the gas stations are placed, there is a starting point for the car such that it can go around the track once (clockwise) without running out of gas.

Problem 3.

Consider the program given below.

```
Function FAC(n)

i <- 1
F <- 1
while i < = n
   F <- F * i
   i <- i + 1
return F
```

Assume that the input to the function, n, is an integer ≥ 0.

A) What mathematical function is computed by this program?

B) After the program goes through the loop k times, what value does F have?

C) Prove your answer to (B).

D) Why is your proof in C) sufficient to prove that the program computes the function given in (A)?

Problem 4. Winning the Game of Nim

The game of Nim has many variations. One simple version is the following: A positive number of sticks are placed on a table. Two players take turns removing one, two, or three sticks. The player to remove the last stick loses.

Claim. The first player has a winning strategy if and only if the number of sticks, n, is *not* 4k + 1 for any k ∈ N.

[Note: A strategy is a rule for how many sticks to remove when there are certain number remaining.]

You can show that if n = 4k+1, then player 2 has a strategy that will force a win, otherwise, player 1 has a strategy that will force a win.

Use induction to prove this.