ECE725/CS745 Winter 2011
Homework 2 (Theorem Proving)

Let $f : X \longrightarrow X$ be a function. In mathematics, a *fixpoint* of function f is $x \in X$, such that $f(x) = x$. For example, if f is defined on the real numbers by

$$f(x) = x^2 - 3x + 4,$$

then 2 is a fixpoint of f, because $f(2) = 2$. Obviously, not all functions have fixpoints. For instance, function $f(x) = x + 1$ on real numbers does not have a fixpoint.

Problem Description
Fixpoint calculation on sets and predicates have direct application in verification. For example, consider the automaton in Figure 1. Using fixpoint calculation, one can compute the set of reachable states from a set of states by applying the transition relation. Once a fixpoint is reached, the set of all reachable states is computed. For instance, the steps to compute the set of reachable states by starting from state s_1 is the following:

1. $\{s_1\}$
2. $\{s_1, s_4\}$
3. $\{s_1, s_4, s_2, s_5\}$
4. $\{s_1, s_4, s_2, s_5, s_3\}$
5. $\{s_1, s_4, s_2, s_5, s_3, s_0\}$

In this example, the application function is increasing (i.e., set union) and the goal is to compute the *smallest* fixpoint. Observe that the fixpoint happnes to be the set of all states.

In this assignment, you are to develop a *largest fixpoint theory* on *finite sets* in PVS and prove its well-known properties as follows. Let g be an abstract function defined as follows:

$$g : \{X : \mathit{finiteset}\} \longrightarrow \{Y : \mathit{finiteset} \mid Y \subseteq X\}$$

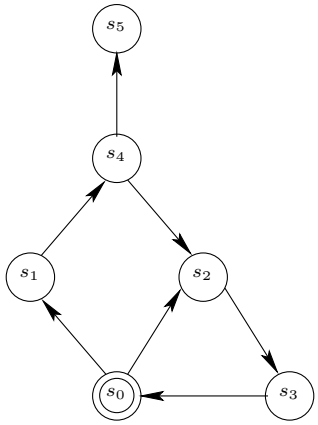

Figure 1:

In other words, $g(x) \subseteq x$. Obviously, g is a decreasing function and since it is abstract, it can be any such function. You will need to define a recursive function D in PVS that given an initial finite set x and a decreasing function g, formalizes the process of computing the fixpoint.

Some hints for specification:

- Let the application function on which we compute the fixpoint be set subtraction.
- You are welcome to use PVS's set induction theory, but it is easirer to use an integer that shows the induction step number. E.g.,

$$D(i, x)(g) = D(i-1, x)(g) - g(D(i-1, x)(g)), \text{ if } i \neq 0.$$

- Then, the fixpoint is a finite set *LgFix* whose elements are present in all steps of the above recursive computation.

Goal:

- **Theorem 1:** You need to show that applying function g on the fixpoint results in the empty set.
- **Theorem 2:** Finally, you should show that the fixpoint is indeed a fixpoint (i.e., further application of function D on the fixpoint results in obtaining the fixpoint). Notice that this theorem is almost trivial, given the correctness of Theorem 1.

Some hints to prove the theorems:

- Define intermediate lemmas. Do not try to prove the theorems as they are. For instance, you need to show that each step of the recursive function indeed reduces the size of the given set.
- You can effectively use set cardinalities in many cases.
- Do not foget to discharge all proof obligations.

Deliverables
You are expected to submit a .pvs and a .prf file through email by **8:30am on Thursday February 3**. Your PVS specification must be fully commented and type checked.