Exercises in Predicate Logic

Gordon J. Pace

Although we have not covered concepts such as numbers, functions and relations in the course, we will be using some of these tools as parts of predicates, to evaluate how predicate logic can be used to reason about objects in computing.

1. In this question, we will look at arrays of numbers. Given an array $A$, we will assume we have a function $\text{size}(A)$ which gives us the number of items in array $A$. The function $\text{item}(A, n)$ gives us the item in array $A$ at position $n$ (with $0 \leq n < \text{size}(A)$). We will define a number of operators based on these functions and predicate logic. For example, the predicate $\text{haszero}(A)$ which is true if the number zero is an item in array $A$: $\text{haszero}(A) \equiv \exists n : \mathbb{N} \cdot 0 \leq n < \text{size}(A) \land \text{item}(A, n) = 0$.

   Similarly, $\text{positive}(A)$ can be defined to be true if and only if all items in the array are positive: $\text{positive}(A) \equiv \forall n : \mathbb{N} \cdot (0 \leq n \land n < \text{size}(A)) \Rightarrow \text{item}(A, n) > 0$.

   (a) Define $\text{empty}(A)$, which is true only when $A$ is the empty array (contains no items).
   (b) Define $\text{zero}(A)$, which is true only when all items in $A$ are zero.
   (c) Define $\text{numbers}(A)$, which is true only when in array $A$, the item at position $n$ is in fact the number $n$.
   (d) Define $\text{constant}(A)$, which is true only when all items in $A$ are equal to each other.
   (e) Define $\text{seq}(A)$, which is true if and only if $A$ contains two sequential items which are the same.
   (f) Define $\text{elt}(A, x)$, which is true if $x$ appears as an item in array $A$.
   (g) Define $\text{cap}(A, n)$, which is true if and only if $n$ is greater than or equal to all items in array $A$.
   (h) Define $\text{max}(A, n)$, which is true if and only if $n$ is the largest item in array $A$.
   (i) Define $\text{equal}(A, B)$, which is true if and only if arrays $A$ and $B$ are equal (same length, same items, same order).
   (j) Define $\text{sorted}(A)$, which is true only when the items in $A$ are in sorted order.

2. In this question we will formalise concepts related to a multi-user operating system. We will assume we have a type USER of all valid usernames and a type RIGHT of all rights a user may have. The predicate $\text{activated}(u)$ is true if and only if $u$ is username activated on the system. The predicate $\text{admin}(u)$ (where $u$ is a USER) will mean that $u$ is an administrator of the system, while $\text{normal}(u)$ means that user $u$ is a normal user. Finally, the predicate $\text{hasright}(u, r)$ is true exactly when user $u$ has right $r$. For example, if we were to write that there is at least one activated administrator, we would write it as: $\exists u : \text{USER} \cdot \text{activated}(u) \land \text{admin}(u)$. Similarly, write the following properties as predicates.

   (a) Every activated user is either an administrator or a normal user.
   (b) No user is both an administrator and a normal user.
   (c) Every administrator has the right CREATEUSER.
   (d) Normal users do not have the right CREATEUSER.
   (e) At least one administrator has all rights.
   (f) All administrators have the same, if not more rights than any normal user.
   (g) All normal users have the same rights.