Sample Solutions
Homework XIII

Problem 1.
The axioms for subtraction are analogous to those for addition and are expressed inductively. Namely,

sub(m,0) = m
sub(m,succ(n)) = pred(sum(m,n)) {i.e., m-(n+1) = (m-n)-1}
sub(m,pred(n)) = succ(sub(m,n)) {i.e., m-(n-1)) = (m-n)+1).

To justify the usual rules of arithmetic, first note that every integer is (equivalent to) either $succ^n(0)$ or $pred^n(0)$ for $n \geq 0$. Then, for example, sub($succ^2$(0),$succ^2$(0)) = pred(sub($succ^2$(0),succ(0))) = $pred^2$(sub($succ^2$(0),0)) = $pred^2$($succ^2$(0)) = pred(succ(0)) = 0, by applying the equations above and those given in the problem. In general, arguments rely on induction.

Problem 2.
Again, we formulate the equations, and carry out our deductions, thinking in terms of the "canonical forms" — every nonerror list is expressed as (equivalent to) a term of the form cons(i_1, cons(i_2, … cons(i_n, null) …). For the drop function we add equations that allow this function to be eliminated from any term, namely

drop(0,s) = s
drop(n,null) = errorList, if n>0
drop(n, cons(i,s)) = drop(n-1, s), if n>0 and i≠errorItem.

For the front function, the approach is similar, namely

front(cons(i,s)) = i, if s≠errorList
front(errorList) = errorItem.

Since he equations for drop permit its elimination, no equations involving both drop and front are necessary. From the understanding of the canonical forms, it is clear that the given equations accomplish the desired behavior. Also, for the null list, we have expressed the error outcome (assuming "errors propagate") — note the importance of forbidding error values in "normal operations" as without this we could conclude e.g., i = front(cons(i,errorList)) = front(errorList) = errorItem for every item i.

Problem 3.
There are a variety of ways to correctly formulate an equivalent BNF, but the most direct is just to "invert" the signatures of the functions, namely

Nat → 0
Nat → s(Nat)
Nat → sum(Nat,Nat)
Nat → top(Stack)
Stack → empty
Stack → push(Stack,Nat)
Stack → pop(Stack).