1994 Paper 2 Question 8

Suppose that *take* and *drop* are ML functions such that *take*(n, s) returns the first n elements of the list s, while *drop*(n, s) returns all but the first n elements of s. Let *length*(s) be the function to compute the length of the list s. Consider the following ML function

```
fun front s = take(length s div 2, s);
fun back s  = drop(length s div 2, s);

fun bsum [ ]    = 0.0
  | bsum [x]    = x
  | bsum s      = bsum front s + bsum back s;

fun sum [ ]     = 0.0
  | sum (x::s) = x + sum s;
```

Give a formal proof that *sum*(*front* s) + *sum*(*back* s) = *sum*(s) for all lists s, explaining what properties of arithmetic you are assuming. [9 marks]

Describe a proof of *bsum*(s) = *sum*(s) for all s using the lemma that you have just established. Do not give a detailed proof but instead outline the main argument. State any additional lemmas required and indicate how they might be proved. [6 marks]

Does proving *bsum*(s) = *sum*(s) for all s in this way ensure that *bsum* and *sum* are completely interchangable in ML programs? Discuss. [5 marks]