# Concrete Subtyping

The downcasting operation that would allow us to safely use references that expect a smaller heap in a context where a larger heap is available requires some preliminary work.

First of all let us understand what kind of operation we will perform whenever we try to use a heap that is too large with respect to the get or set functions of the reference:

1. We downcast the larger heap to an adequately smaller one
2. We perform our computation on the downcast heap, thereby obtaining a new smaller heap
3. We store the new smaller heap in the corresponding locations of the original heap

The three steps listed above are summarized in the following diagram:

**a. Downcast**

**b. Operation**

**c. Merge  
back**

The downcast operation is implemented easily enough on heaps thanks to the transitivity of the subtyping relationship. We just need to specify that:

And the first step of the computation is covered. Now we define the in-place substitution operation with an appropriate predicate:

This new predicate is instanced inductively on the length of the prefix:

and

Thanks to this new operator, which we could consider an upcasting operator of sorts, we can now define the proper downcasting operation for references:

This operation invokes the get and set functions of a reference with the downcast (smaller) heap with respect to the input (larger) heap, and then rebuilds a larger heap by stitching together the input heap with the resulting heap from the get or set operation.