

## Generalizing array diagrams

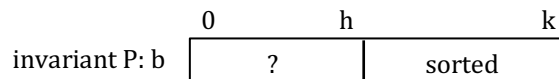
When the precondition and postcondition are given by array diagrams, the invariant is usually a generalization of both of them. Here is the general guideline for creating that generalization.

**Guideline: The segments in R appear in the same order in the generalization.  
The segments in Q appear in the same order in the generalization.**

Here's an example:



Generalizing the two is easy. We have placed a variable  $h$  in the diagram to help mark the boundary between the two segments. You often have to do that.

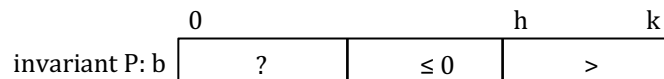


There is another way to generalize  $Q$  and  $R$ . Stop the video and draw the other generalization. The answer appears at the end of the pdf script for this video.

Here's a second example of a pre- and post-condition:

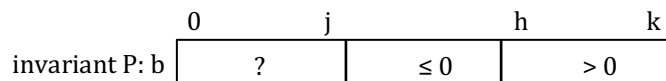


To generalize these, we can start with the postcondition and place the query segment from  $Q$  at the beginning:



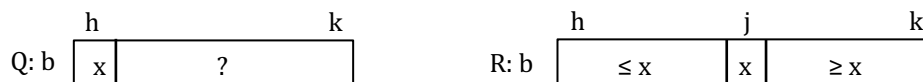
Note that the segments in  $R$  appear in the same order in the invariant. Variable  $h$  appears in the same place in both the postcondition and the invariant.

In order to use this invariant, we have to mark the boundary between the first two segments. We can put a variable  $j$  like this. We could also have placed  $j$  to the right of the line. As you gain experience developing the loops, you will learn which one may be (slightly) better.

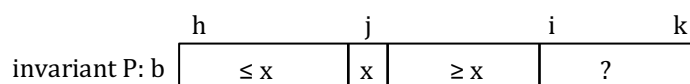


There are two other ways to generalize  $Q$  and  $R$ , depending on where the query segment is placed. Stop the video and draw those two ways. The answers appear at the end of the pdf script for this video.

Our last example comes from a problem in which  $b[h]$  contains some value  $x$ ; the values in  $b[h..k]$  are to be swapped around and a value is to be stored in  $j$  to truthify the postcondition. But we don't concern ourselves with that! We just generalize  $Q$  and  $R$ .



To generalize, we start with  $R$ , notice that we don't need to put a segment with  $x$  in it because there is one in  $R$ , and decide to place the query segment at the end. We use fresh variable  $i$  to mark the boundary.



Note that in all three diagrams,  $h$  and  $k$  give the beginning and end of the array. Variable  $j$  is used in the invariant as it is used in the postconditions.

There is another way to generalize, depending on where the query segment is placed. Draw that one.

## Generalizing array diagrams

### Answers

1. The other generalization in the first example is:

	0	h	k
invariant P: b	sorted	?	

2. The two other generalizaations in the second example are:

	0	j	h	k
invariant P: b	$\leq 0$	?	$> 0$	

	0	h	j	k
invariant P: b	$\leq 0$	$> 0$	?	

3. The other generalization in the third example is:

	h	j	i	k
Invariant P: b	$\leq x$	x	?	$\geq x$