

467 A binary tree can be stored as a list of nodes in breadth order. Traditionally, the root is at index 1, the node at index  $n$  has its left child at index  $2n$  and its right child at index  $2n+1$ . Suppose the user's variable is  $x: X$ , and the implementer's variables are  $s: [*X]$  and  $p: nat+1$ , and the operations are

<i>goHome</i>	=	$p := 1$
<i>goLeft</i>	=	$p := 2 \times p$
<i>goRight</i>	=	$p := 2 \times p + 1$
<i>goUp</i>	=	$p := \text{div } p \ 2$
<i>put</i>	=	$s := p \rightarrow x \mid s$
<i>get</i>	=	$x := s \ p$

Now suppose we decide to move the entire list down one index so that we do not waste index 0. The root is at index 0, its children are at indexes 1 and 2, and so on. Develop the necessary data transformer, and use it to transform the operations.

After trying the question, scroll down to the solution.

§ The new implementer's variables are  $r: [*X]$  and  $q: nat$ . The transformer is

$$r = s[1;..#s] \wedge p = q+1$$

Program  $P$  is transformed to

$$\forall s, p. r = s[1;..#s] \wedge p = q+1 \Rightarrow \exists s', p'. r' = s'[1;..#s'] \wedge p' = q'+1 \wedge P$$

It is easy enough to eliminate the three variables  $p$ ,  $s'$ , and  $p'$  by one-point. The problem is to get rid of  $s$  because we don't have  $s=\text{something}$ . We have

$$r = s[1;..#s]$$

From this we see that  $s = [i];;r$  for some unknown item  $i$ . I'll use that to eliminate  $s$ . This is probably the Change of Variable law, but I don't see the match at the moment, so I'll fudge a bit with item  $i$ . Program  $P$  is transformed to

$$\forall s, p. s = [i];;r \wedge p = q+1 \Rightarrow \exists s', p'. r' = s'[1;..#s'] \wedge p' = q'+1 \wedge P$$

Transform  $goHome$ :

$$\begin{aligned} & \forall s, p. s = [i];;r \wedge p = q+1 \Rightarrow \exists s', p'. r' = s'[1;..#s'] \wedge p' = q'+1 \wedge goHome \\ = & \forall s, p. \quad s = [i];;r \wedge p = q+1 \\ & \Rightarrow \exists s', p'. r' = s'[1;..#s'] \wedge p' = q'+1 \wedge x'=x \wedge s'=s \wedge p'=1 \\ & \hspace{15em} \text{One-point 4 times} \\ = & r' = ([i];;r)[1;..#[i];;r] \wedge 1 = q'+1 \wedge x'=x \\ & \hspace{15em} \text{The list } [i];;r \text{ is indexed from 1 to its end.} \\ = & r'=r \wedge q'=0 \wedge x'=x \\ = & q:=0 \end{aligned}$$

Transform  $goLeft$ :

$$\begin{aligned} & \forall s, p. s = [i];;r \wedge p = q+1 \Rightarrow \exists s', p'. r' = s'[1;..#s'] \wedge p' = q'+1 \wedge goLeft \\ = & \forall s, p. \quad s = [i];;r \wedge p = q+1 \\ & \Rightarrow \exists s', p'. r' = s'[1;..#s'] \wedge p' = q'+1 \wedge x'=x \wedge s'=s \wedge p'=2 \times p \\ & \hspace{15em} \text{One-point 4 times} \\ = & r' = ([i];;r)[1;..#[i];;r] \wedge 2 \times (q+1) = q'+1 \wedge x'=x \\ & \hspace{15em} \text{The list } [i];;r \text{ is indexed from 1 to its end.} \\ = & r'=r \wedge q' = 2 \times q + 1 \wedge x'=x \\ = & q:= 2 \times q + 1 \end{aligned}$$

Transform  $goRight$ :

$$\begin{aligned} & \forall s, p. s = [i];;r \wedge p = q+1 \Rightarrow \exists s', p'. r' = s'[1;..#s'] \wedge p' = q'+1 \wedge goRight \\ = & \forall s, p. \quad s = [i];;r \wedge p = q+1 \\ & \Rightarrow \exists s', p'. r' = s'[1;..#s'] \wedge p' = q'+1 \wedge x'=x \wedge s'=s \wedge p'=2 \times p + 1 \\ & \hspace{15em} \text{One-point 4 times} \\ = & r' = ([i];;r)[1;..#[i];;r] \wedge 2 \times (q+1) + 1 = q'+1 \wedge x'=x \\ & \hspace{15em} \text{The list } [i];;r \text{ is indexed from 1 to its end.} \\ = & r'=r \wedge q' = 2 \times q + 2 \wedge x'=x \\ = & q:= 2 \times q + 2 \end{aligned}$$

Transform  $goUp$ :

$$\begin{aligned} & \forall s, p. s = [i];;r \wedge p = q+1 \Rightarrow \exists s', p'. r' = s'[1;..#s'] \wedge p' = q'+1 \wedge goUp \\ = & \forall s, p. \quad s = [i];;r \wedge p = q+1 \\ & \Rightarrow \exists s', p'. r' = s'[1;..#s'] \wedge p' = q'+1 \wedge x'=x \wedge s'=s \wedge p' = \text{div } p \ 2 \\ & \hspace{15em} \text{One-point 4 times} \\ = & r' = ([i];;r)[1;..#[i];;r] \wedge \text{div } (q+1) \ 2 = q'+1 \wedge x'=x \\ & \hspace{15em} \text{The list } [i];;r \text{ is indexed from 1 to its end.} \\ = & r'=r \wedge q' = \text{div } (q+1) \ 2 - 1 \wedge x'=x \\ = & q:= \text{div } (q+1) \ 2 - 1 \\ = & q:= \text{div } (q-1) \ 2 \end{aligned}$$

Transform *put* :

$$\begin{aligned}
& \forall s, p. s = [i];;r \wedge p = q+1 \Rightarrow \exists s', p'. r' = s'[1;..\#s'] \wedge p' = q'+1 \wedge \textit{put} \\
= & \forall s, p. s = [i];;r \wedge p = q+1 \\
& \Rightarrow \exists s', p'. r' = s'[1;..\#s'] \wedge p' = q'+1 \wedge x'=x \wedge s'=p \rightarrow x \mid s \wedge p'=p \\
& \text{One-point 4 times} \\
= & r' = (q+1 \rightarrow x \mid [i];;r)[1;..\#(q+1 \rightarrow x \mid [i];;r)] \wedge q+1 = q'+1 \wedge x'=x \\
& \text{We can simplify } \#(q+1 \rightarrow x \mid [i];;r) \text{ ro } \#r+1 \text{ and simplify } q+1 = q'+1 \text{ to } q'=q. \\
= & r' = (q+1 \rightarrow x \mid [i];;r)[1;..\#r+1] \wedge q'=q \wedge x'=x \\
& \text{We have a list } (q+1 \rightarrow x \mid [i];;r) \text{ of length } \#r+1, \\
& \text{and in this list at index } q+1 \text{ the item is } x. \\
& \text{Now we index with the list } [1;..\#r+1], \\
& \text{which shifts all the indexes down } 1. \\
& \text{So now at index } q \text{ the item is } x. \\
= & r' = q \rightarrow x \mid r \wedge q'=q \wedge x'=x \\
= & r := q \rightarrow x \mid r
\end{aligned}$$

Transform *get* :

$$\begin{aligned}
& \forall s, p. s = [i];;r \wedge p = q+1 \Rightarrow \exists s', p'. r' = s'[1;..\#s'] \wedge p' = q'+1 \wedge \textit{get} \\
= & \forall s, p. s = [i];;r \wedge p = q+1 \\
& \Rightarrow \exists s', p'. r' = s'[1;..\#s'] \wedge p' = q'+1 \wedge x'=s p \wedge s'=s \wedge p'=p \\
& \text{One-point 4 times} \\
= & r' = ([i];;r)[1;..\#[[i];;r]] \wedge q+1 = q'+1 \wedge x' = ([i];;r)(q+1) \\
& \text{The list } [i];;r \text{ is indexed from } 1 \text{ to its end.} \\
& \text{And } ([i];;r)(q+1) = r q \\
= & r'=r \wedge q'=q \wedge x'=r q \\
= & x := r q
\end{aligned}$$