



**Exercise 5.** For each heap implication below, say whether it is true or false.

1.  $(r \mapsto 3) * (s \mapsto 4) \triangleright (s \mapsto 4) * (r \mapsto 3)$
2.  $(r \mapsto 3) \triangleright (s \mapsto 4) * (r \mapsto 3)$
3.  $(s \mapsto 4) * (r \mapsto 3) \triangleright (r \mapsto 4)$
4.  $(s \mapsto 4) * (r \mapsto 3) \triangleright (r \mapsto 3)$
5.  $\text{'False'} * (r \mapsto 3) \triangleright (s \mapsto 4) * (r \mapsto 4)$
6.  $(r \mapsto 4) * (s \mapsto 3) \triangleright \text{'False'}$
7.  $(r \mapsto 4) * (r \mapsto 3) \triangleright \text{'False'}$
8.  $(r \mapsto 3) * (r \mapsto 3) \triangleright \text{'False'}$

**Exercise 6.** For each heap implication below, say whether it is true or false.

1.  $(r \mapsto 3) \triangleright \exists n. (r \mapsto n)$
2.  $\exists n. (r \mapsto n) \triangleright (r \mapsto 3)$
3.  $\exists n. (r \mapsto n) * \text{'}n > 0\text{'} \triangleright \exists n. \text{'}n > 1\text{'} * (r \mapsto (n - 1))$
4.  $(r \mapsto 3) * (s \mapsto 3) \triangleright \exists n. (r \mapsto n) * (s \mapsto n)$
5.  $\exists n. (r \mapsto n) * \text{'}n > 0\text{'} * \text{'}n < 0\text{'} \triangleright (r \mapsto n) * (r \mapsto n)$

**Exercise 7.** show that GC-PRE is derivable from GC-POST and FRAME.

$$\frac{\{H\} t \{Q\}}{\{H * \text{GC}\} t \{Q\}}$$

**Exercise 8.** give a specification of copy in terms of MtreeComplete; which rules are used to derive this specification?

**Exercise 9.** complete the rule for sequences.

$$\frac{\{ \quad \} t_1 \{ \quad \} \quad \{ \quad \} t_2 \{ \quad \}}{\{H\} (t_1 ; t_2) \{Q\}}$$

**Exercise 10.** complete the reasoning rule for let-bindings.

$$\frac{\{ \quad \} t_1 \{ \quad \} \quad \forall x. \{ \quad \} t_2 \{ \quad \}}{\{H\} (\text{let } x = t_1 \text{ in } t_2) \{Q\}}$$

**Exercise 11.** instantiate the rule for let-bindings on the following code.

$$\{r \mapsto 3\} (\text{let } a = !r \text{ in } a+1) \{Q\}$$

$$H \equiv$$

$$Q \equiv$$

$$Q' \equiv$$

**Exercise 12.** Reasoning rule for values:

$$\frac{\triangleright}{\{H\} v \{Q\}}$$

**Exercise 13.** give a proof sketch for the induction for `mLength`.

**Exercise 14.** give two specifications for the function `refapply`.

In the first, assume `f` to be pure, and introduce a predicate  $P x y$ .

In the second, assume that `f` also modifies the state from  $H$  to  $H'$ .

$$\left\{ \begin{array}{l} \forall r f x H H' P. \{ \quad \} (f x) \{ \lambda . \quad \} \\ \{ \quad \} (\text{refapply } r f) \{ \lambda . \quad \} \end{array} \right\} \Rightarrow \left\{ \quad \right\}$$

**Exercise 15.** specify `repeat`, using an invariant  $I$ , of type  $\text{int} \rightarrow \text{Hprop}$ .

**Exercise 16.** specify `iter`, using an invariant  $I$ , of type  $\text{list } \alpha \rightarrow \text{Hprop}$ .

**Exercise 17.** Give an example where ITER-PAST is insufficient

**Exercise 18.** Show ITER-FUTURE implies ITER-PAST