

CSE 331 HW3

Rohan Mukherjee

July 16, 2024

1. (b) Since the patterns are fixed to be 4×2 , and there is only one color option, there is only one test case per function, i.e. correctness level 0. This is < 20 so we can just test all of them and look to see that they look right.
- (d) Since the patterns are still 4×2 , except now we have a color option, each function only has two inputs, i.e. correctness level 0. So we can just test all of them and see that they look right.
- (f) Since these functions were straight from the spec, it has correctness level 1. We need to use our testing heuristics, in this case since all the functions are recursive, we need to use the 0-1-many heuristic to test the functions are correct. It is important to write the function as closely to the mathematical description as possible to ensure that we don't need to do reasoning, instead just testing.

2. (a)

$$\begin{aligned} \text{sum}(\text{cons}(a, \text{cons}(b, \text{nil}))) &= a + \text{sum}(\text{cons}(b, \text{nil})) && \text{def of sum} \\ &= a + b + \text{sum}(\text{nil}) && \text{def of sum} \\ &= a + b && \text{def of sum} \end{aligned}$$

(b)

$$\begin{aligned} \text{sum}(\text{twice} - \text{evens}(\text{cons}(a, \text{cons}(b, \text{nil})))) &= \text{sum}(\text{cons}(2a, \text{cons}(b, \text{twice} - \text{evens}(\text{nil})))) && \text{def of t-e} \\ &= 2a + \text{sum}(\text{cons}(b, \text{twice} - \text{evens}(\text{nil}))) && \text{def of sum} \\ &= 2a + \text{sum}(\text{cons}(b, \text{nil})) && \text{def of t-e} \\ &= 2a + b && \text{def of sum} \end{aligned}$$

(c)

$$\begin{aligned} \text{sum}(\text{twice} - \text{odds}(\text{cons}(a, \text{cons}(b, \text{nil})))) &= \text{sum}(\text{cons}(a, \text{cons}(2b, \text{twice} - \text{odds}(\text{nil})))) && \text{def of t-o} \\ &= a + \text{sum}(\text{cons}(2b, \text{twice} - \text{odds}(\text{nil}))) && \text{def of sum} \\ &= a + \text{sum}(\text{cons}(2b, \text{nil})) && \text{def of t-o} \\ &= a + 2b && \text{def of sum} \end{aligned}$$

(d)

$$\begin{aligned} \text{sum}(\text{twice} - \text{evens}(L)) + \text{sum}(\text{twice} - \text{odds}(L)) &= 2a + b + \text{sum}(\text{twice} - \text{odds}(L)) && \text{part (b)} \\ &= 2a + b + a + 2b && \text{part (c)} \\ &= 3a + 3b \\ &= 3(a + b) \\ &= 3 \cdot \text{sum}(L) && \text{part (a)} \end{aligned}$$

3. (a) The mathematical definition for flipping a single block is as follows:

$$\begin{aligned} \text{bflip-vert}(\{\text{design: } d, \text{color: } c, \text{corner: NE}\}) &= \{\text{design: } d, \text{color: } c, \text{corner: SE}\} \\ \text{bflip-vert}(\{\text{design: } d, \text{color: } c, \text{corner: NW}\}) &= \{\text{design: } d, \text{color: } c, \text{corner: SW}\} \\ \text{bflip-vert}(\{\text{design: } d, \text{color: } c, \text{corner: SE}\}) &= \{\text{design: } d, \text{color: } c, \text{corner: NE}\} \\ \text{bflip-vert}(\{\text{design: } d, \text{color: } c, \text{corner: SW}\}) &= \{\text{design: } d, \text{color: } c, \text{corner: NW}\} \end{aligned}$$

(c)

$$\begin{aligned} rnil & & rnil \\ rcons(a, rnil) & & rcons(b(a), rnil) \\ rcons(a, rcons(c, rnil)) & & rcons(b(a), rcons(b(c), rnil)) \\ rcons(a, rcons(c, rcons(d, rnil))) & & rcons(b(a), rcons(b(c), rcons(b(d), rnil))) \end{aligned}$$

(d)

$$\begin{aligned} \text{rflip-vert}(rnil) &= rnil \\ \text{rflip-vert}(rcons(a, R)) &= rcons(b(a), \text{rflip-vert}(R)) \end{aligned}$$

(f)

$$\begin{array}{ll} qnil & qnil \\ qcons(t, qnil) & qcons(r(t), qnil) \\ qcons(t, qcons(v, qnil)) & qcons(r(v), qcons(r(t), qnil)) \\ qcons(t, qcons(v, qcons(w, qnil))) & qcons(r(w), qcons(r(v), qcons(r(t), qnil))) \end{array}$$

(g)

$$\begin{aligned} qflip\text{-}vert(qnil) &= qnil \\ qflip\text{-}vert(qcons(t, Q)) &= qconcat(qflip\text{-}vert(Q), qcons(t, qnil)) \end{aligned}$$

4. (a)

$$\begin{aligned} bflip\text{-}horz(\{\text{design: } d, \text{color: } c, \text{corner: NE}\}) &= \{\text{design: } d, \text{color: } c, \text{corner: NW}\} \\ bflip\text{-}horz(\{\text{design: } d, \text{color: } c, \text{corner: NW}\}) &= \{\text{design: } d, \text{color: } c, \text{corner: NE}\} \\ bflip\text{-}horz(\{\text{design: } d, \text{color: } c, \text{corner: SE}\}) &= \{\text{design: } d, \text{color: } c, \text{corner: SW}\} \\ bflip\text{-}horz(\{\text{design: } d, \text{color: } c, \text{corner: SW}\}) &= \{\text{design: } d, \text{color: } c, \text{corner: SE}\} \end{aligned}$$

(c) From the above, we define:

$$\begin{aligned} rflip\text{-}horz(rnil) &= rnil \\ rflip\text{-}horz(rcons(a, R)) &= rconcat(rflip\text{-}horz(R), rcons(b(a), rnil)) \end{aligned}$$

(e)

$$\begin{aligned} qflip\text{-}horz(qnil) &= qnil \\ qflip\text{-}horz(qcons(t, Q)) &= qcons(r(t), qflip\text{-}horz(Q)) \end{aligned}$$

(h) Since the symmetrize function has already been tested, we only need to test that inputting the symmetrize tag into the URL is working properly, and symmetrizes the image if it appears, and not otherwise. This was an imperative spec (it told us exactly what to do), and has only a small number of inputs, namely if symmetrize is present or not, so we can just test all of them. Putting URLs with/without the symmetrize tag and checking if the image is symmetrized or not is sufficient.

5. (a)

$$\begin{aligned} swap(swap(cons(x, nil))) &= swap(cons(x, nil)) && \text{def of swap} \\ &= cons(x, nil) && \text{def of swap} \end{aligned}$$

(b)

$$\begin{aligned} \text{swap}(\text{swap}(\text{cons}(x, \text{cons}(y, R)))) &= \text{swap}(\text{cons}(y, \text{cons}(x, \text{swap}(R)))) && \text{def of swap} \\ &= \text{cons}(x, \text{cons}(y, \text{swap}(\text{swap}(R)))) && \text{def of swap} \end{aligned}$$

(c) Let $S = \text{cons}(u, \text{cons}(v, \text{cons}(w, \text{nil})))$. Then,

$$\begin{aligned} \text{swap}(\text{swap}(\text{cons}(s, \text{cons}(t, S)))) &= \text{cons}(s, \text{cons}(t, \text{swap}(\text{swap}(S)))) && \text{part (b)} \\ &= \text{cons}(s, \text{cons}(t, \text{cons}(u, \text{cons}(v, \text{swap}(\text{swap}(w, \text{nil})))))) && \text{part (b)} \\ &= \text{cons}(s, \text{cons}(t, \text{cons}(u, \text{cons}(v, \text{cons}(w, \text{nil})))))) && \text{part (a)} \end{aligned}$$

(d) If L has length 3, then we can see that

$$\begin{aligned} \text{len}(\text{cons}(1, \text{cons}(2, L))) &= 1 + \text{len}(\text{cons}(2, L)) && \text{def of len} \\ &= 1 + 1 + \text{len}(L) && \text{def of len} \\ &= 2 + \text{len}(L) && \text{def of len} \\ &= 5 \end{aligned}$$

So, we can apply part (c) to see that,

$$\text{swap}(\text{swap}(\text{cons}(1, \text{cons}(2, L)))) = \text{cons}(1, \text{cons}(2, L)) \quad \text{part (c)}$$

6. (a) Since we know that $\text{cons}(a, L) \neq \text{nil}$, we use the definition of cycle to see that $\text{cycle}(\text{cons}(a, L)) = \text{concat}(L, \text{cons}(a, \text{nil}))$. Now, if L is nil then $\text{concat}(L, \text{cons}(a, \text{nil})) = \text{cons}(a, \text{nil})$ is not nil. Otherwise, write $L = \text{cons}(b, M)$, then $\text{concat}(L, \text{cons}(a, \text{nil})) = \text{cons}(b, \text{concat}(M, \text{cons}(a, \text{nil})))$ by definition of concat is not nil since it at least contains b .

(b) From part (a), we know that $\text{cycle}(\text{cons}(a, L)) \neq \text{nil}$. Thus we can write $\text{cycle}(\text{cons}(a, L)) = \text{cons}(b, R)$ for some $b \in \mathbb{Z}$ and some list R . by definition of len, we have that

$$\text{len}(\text{cons}(b, R)) = 1 + \text{len}(R)$$

Now, per the hint we know that $\text{len}(R) \geq 0$. Thus, $\text{len}(\text{cycle}(\text{cons}(a, L))) = \text{len}(\text{cons}(b, R)) = 1 + \text{len}(R) \geq 1$, and we are done.