

## A8: Induction

Due: April 8th, 2020

Please refer to the lecture notes and the accompanying Java code for all definitions.

1.  $xs.map(id) = xs$
2.  $(xs.append(ys)).map(f) = (xs.map(f)).append(ys.map(f))$
3.  $(xs.append(ys)).fold(g, a) = xs.fold(g, ys.fold(g, a))$
4.  $(xs.append(ys)).length() = xs.length() + ys.length()$
5.  $(xs.reverseH(ys)).length() = xs.length() + ys.length()$
6.  $xs.length() = (xs.reverse2()).length()$
7.  $(xs.append(ys)).reverse() = ys.reverse().append(xs.reverse())$
8.  $(xs.reverse()).reverse() = xs$
9.  $xs.reverseH(ys) = (xs.reverse()).append(ys)$
10.  $(xs.reverseH(ys)).reverseH(zs) = ys.reverseH(xs.append(zs))$
11.  $xs.reverseH(ys.append(zs)) = (xs.reverseH(ys)).append(zs)$
12.  $(xs.append(ys)).reverseH(zs) = ys.reverseH(xs.reverseH(zs))$
13.  $(xs.append(ys)).reverse2() = ys.reverse2().append(xs.reverse2())$
14.  $(xs.reverse2()).reverse2() = xs$
15.  $t.flattenH(xs) = t.flatten().append(xs)$
16.  $t.flatten2() = t.flatten()$
17.  $t.map(f1).sum() = t.nodes()$
18.  $t.nodes() = t.longestPath().length() + 1$ .
19. For non-empty trees  $t$ , it is the case that  $t.internalNodes() + 1 = t.leaves()$ .
20. A full  $m$ -ary with  $n$  nodes has  $(n - 1)/m$  internal nodes and  $((m - 1)n + 1)/m$  leaves.
21. A full  $m$ -ary with  $i$  internal nodes has  $mi + 1$  nodes and  $(m - 1)i + 1$  leaves.
22. A full  $m$ -ary with  $l$  leaves has  $(ml - 1)/(m - 1)$  nodes and  $(l - 1)/(m - 1)$  internal nodes.
23. How many people have seen the letter, including the first person?
24. How many people sent out the letter?