Solution to Problem 12

Problem. In class, we described a Turing machine that computes $g(n) = n+1$. In Homework 9, you designed a Turing machine that computes a function $f(n)$ which is equal to $n-2$ when $n \geq 2$ and to 0 when $n = 0$.

In class, we described the general algorithm for designing a Turing machine that computes the composition of two functions. The assignment is to use this general algorithm to design a Turing machine that computes the composition $g(f(n))$. Trace, step-by-step, on an example, how your Turing machine works. For example, you can take as input $n = 2$.

Reminder: The Turing machine for computing $g(n) = n + 1$ for a unary input n is based on the following idea:

- we go step-by-step until we find the first blank space,
- then, we replace this blank space with 1 and go back.

This machine has the following rules:

- start, – → R, working
  (we start going to the right)
- working, 1 → R
  (we see 1, so we continue going),
- working, – → 1, L, back
  (we see a blank space, so we replace it with 1 and start going back)
- back, → L
  (while we see 1s, we continue going back)
- back, – → halt
  (once we reach the very first cell, we stop).

The Turing machine for computing $f(n)$ is based on the following idea:

- we go step-by-step until we find the first blank space,
- then, we go back, replace the last two 1s with blanks, and go back all the way.

We need to take special care of the case when $n = 0$.

Solution. The resulting Turing machine takes the following form:

- start, – → R, moving_1
- moving_1, 1 → R
- moving_1, – → erasing1st_1
- erasing1st_1, 1 → –, L, erasing2nd_1
- erasing2nd_1, 1 → –, L, back_1
- back_1, 1 → L
- back_1, – → start_2
- erasing1st_1, – → start_2
- erasing2nd_1, – → start_2
- start_2, – → R, working_2
- working_2, 1 → R
- working_2, – → 1, L, back_2
- back_2, → L
- back_2, – → halt

Let us trace it for $n = 2$;

$\underline{-}$	1	1	–	...	start
–	$\underline{1}$	1	–	...	moving_1
–	1	$\underline{1}$	–	...	moving_1
–	1	1	$\underline{-}$	...	moving_1
–	1	$\underline{1}$	–	...	erasing1st_1
–	$\underline{1}$	–	–	...	erasing2nd_1
$\underline{-}$	–	–	–	...	back_1
$\underline{-}$	–	–	–	...	start_2
–	$\underline{-}$	–	–	...	working_2
$\underline{-}$	1	–	–	...	back_2
$\underline{-}$	1	–	–	...	halt