Models of Computation 2023

Herbert strikes again

Question 1:

Question 2:

**A:**

The length of representation of the list of instructions is the sum of all the instruction's representations, plus one for each divider between them. Therefore, the length of the program in bits is:

9 + 54 + 20 + 16 + 2808 + 7 + 69600 + 30688 + 68 + 0

= 103270 bits

Each instruction can be decoded as follows:

54 = 0b110110

= <<1, 0b1101>>

= <<1, 13>>

= <<1, <1, 0b11>>>

= <<1, <1, 3>>>

= R0- -> L1, L3

20 = 0b10100

= <<2, 0b10>>

= <<2, 2>>

= R1+ -> L2

16 = 0b10000

= <<4, 0b0>>

= <<4, 0>>

= R2+ -> L0

2808 = 0b101011111000

= <<3, 0b10101111>>

= <<3, <4, 0b101>>>

= <<3, <4, 5>>>

= R1- -> L4, L5

7 = 0b111

= <<0, 0b11>>

= <<0, 3>>

= R0+ -> L3

69600 = 0b10000111111100000

= <<5, 0b10000111111>>

= <<5, <6, 0b1000>>>

= <<5, <6, 8>>>

= R2- -> L6, L8

30688 = 0b111011111100000

= <<5, 0b111011111>>

= <<5, <5, 0b111>>>

= <<5, <5, 7>>>

= R2- -> L5, L7

68 = 0b1000100

= <<2, 0b1000>>

= <<2, 8>>

= R1+ -> L8

0 = HALT

The full program is:

L0: R0- -> L1, L3

L1: R1+ -> L2

L2: R2+ -> L0

L3: R1- -> L4, L5

L4: R0+ -> L3

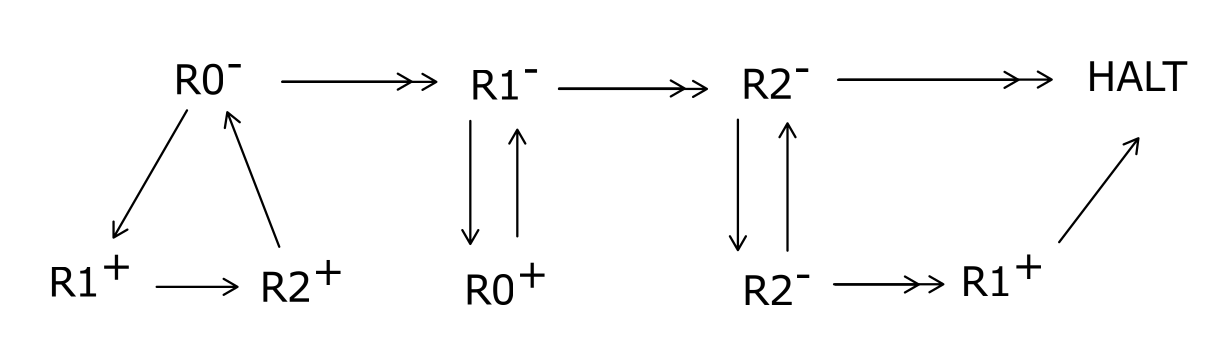
L5: R2- -> L6, L8

L6: R2- -> L5, L7

L7: R1+ -> L8

L8: HALT

This has a graphical representation of:



This program computes whether the value initially in R0 is odd or even. When the program halts, the original value of R0 is in R0, and if that value is odd, R1 holds 1, otherwise it holds 0.

**B:**

*Note to reader: This question has a typo in it that makes it much easier than it was meant to be. In the exam, it was left uncorrected, and marked that way. Expect there to be typos in the future as well.*

We'll describe execution with a tuple of (state, left, right), where left and right are lists of symbols to the left and right of the head, where the symbol the head of the tape is currently over is the first element in right.

We get an execution that looks like this for the tape 000 (\_ represents the empty symbol, and e the empty list):

(q1, e, 000)

(q2, \_, 00)

(q2, \_1, 0)

(q2, \_11, e)

(a, \_11\_, e)

And for the tape 0000:

(q1, e, 0000)

(q2, \_, 000)

(q2, \_1, 00)

(q2, \_11, 0)

(q2, \_111, e)

(a, \_111\_, e)

This Turing machine will accept ***all*** tapes containing only zeros initially.

**C:**

A grounded term is one in which there are no free variables. For example, (\x . y) isn’t grounded, and (\x . x) is.

No, not every grounded term has a normal form under beta reduction, as we can produce counter examples of grounded terms that beta reduce to themselves.

One counter-example is: ((\x . x x)(\x . x x))

Plus is defined as: plus = (\m n f x . m f (n f x))

One is the function that applies a function to a value once: 1 = (\f x . f x)

So plus 1 1 is:

((\m n f x . m f (n f x)) (\f x . f x) (\f x . f x))

β> ((\n f x . (\f x . f x) f (n f x)) (\f x . f x))

β> (\f x . (\f x . f x) f ((\f x . f x) f x))

β> (\f x . (\f x . f x) f (f x))

β> (\f x . f (f x))