Qn 1

We will reduce YesOnEmpty (YoE) to OCI, thus showing OCI is uncomputable, since YoE is known to be uncomputable. Let P be an arbitrary input to YoS. Create a program P′ as follows. Given input string I′, P′ first computes via simulation $v = P(\varepsilon)$. If $v =$ 'yes' then P′ returns I′ (or any string containing I′ would also work). Otherwise, P′ returns 'no'.

By construction, OCI(P′) has a positive solution iff YoE(P) = 'yes'. This completes the reduction and the proof.

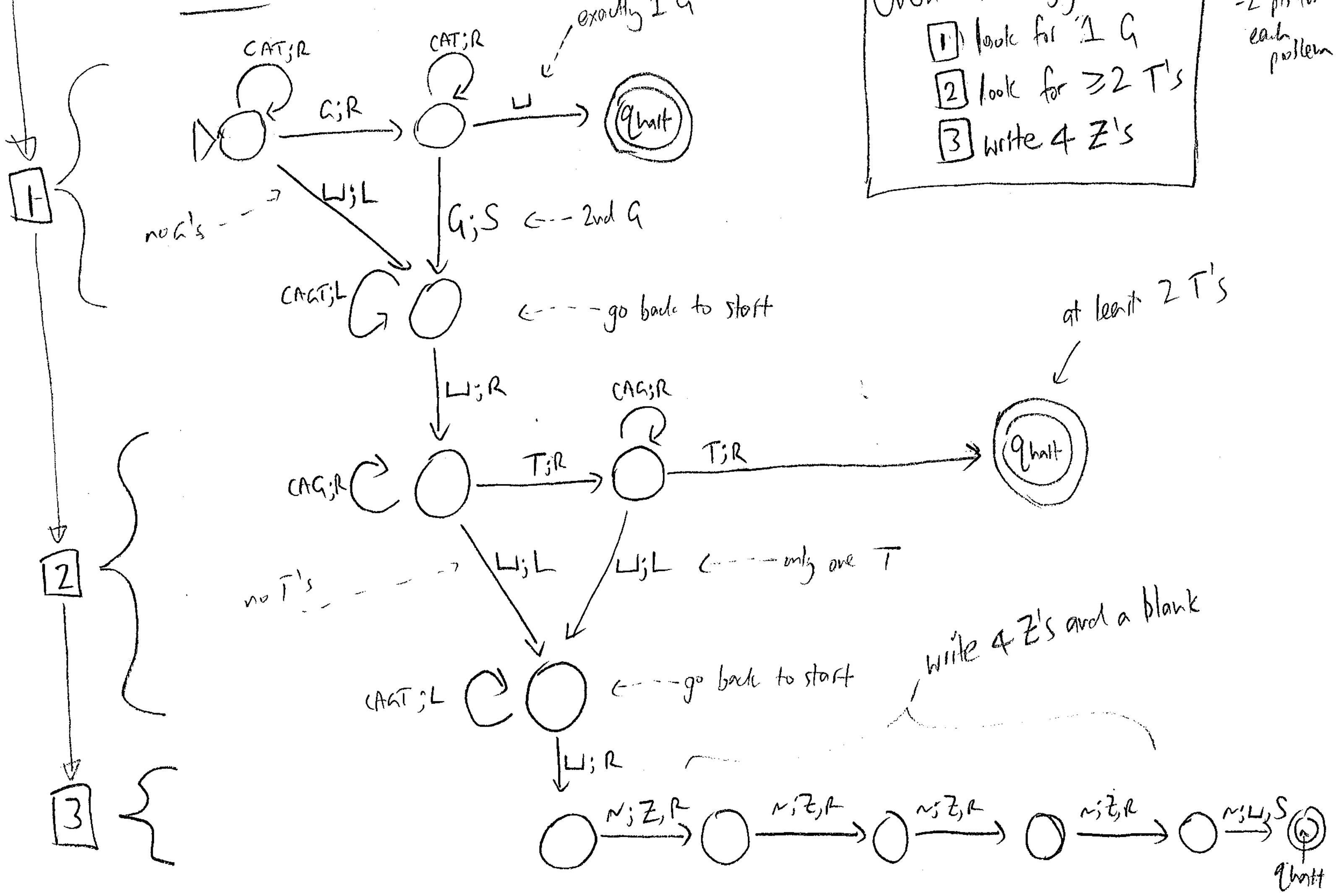
take $S = \{F$ such that $F(I)$ contains I for at least one I with $|I| \geq 3\}$

Qn 1 Note: Can also prove using Rice's theorem or explicit Python programs.

rubric: · G
· TT
· ε
-2 pts for each problem

Qn 2

Overall strategy:
1. look for 1 G
2. look for ≥2 T's
3. write 4 Z's



Qn 3

L is regular if & only if L^R is regular. We show L^R is not regular.

Note $L^R = \{ CCT^{3m} A^{n+2} GG$ such that $n > m \}$

This is an infinite language. Assume it is regular and argue for a contradiction.

By the pumping lemma, there exits a cutoff N such that all strings in L^R that are longer than N can be pumped before N. We may assume N is a multiple of 3. (If not, increase N by 1 or 2.)

Consider

$$S = CCT^N A^{\frac{N}{3}+3} GG$$

Note $S \in L^R$. So some substring w in the first N characters can be pumped.

- If w contains a C, we pump it twice and immediately obtain a contradiction — there are too many C's.
- Otherwise, $w = T^k$ for some $k \geq 1$. Pumping four times we obtain

$$S' = CCT^{N+3k} A^{\frac{N}{3}+3} GG$$

— but S' is not in L^R, since the number of A's is too small —

full details → ... not required for full credit.

writing in the form $T^{3m} A^{n+2}$

we have $N+3k = 3m$ i.e. $m = \frac{N}{3} + k$

and $\frac{N}{3} + 3 = n+2$ i.e. $n = \frac{N}{3} + 1$

so $n \leq m$ (recall $k \geq 1$)

contradicting the requirement that $n > m$.

Qn 4

Yes, TMprintsGin20 is computable. The solution can be computed by simulating the given M for all possible inputs up to length 20. If a G is ever printed, the solution is positive. Otherwise, 'no'.

Qn 5

No. Each quantum core is Turing-equivalent to a classical core. Multiple classical cores are Turing-equivalent to a single core. Thus, the multicore quantum machine can solve the same problems as a classical computer.