A solution) A reg m/c has a FINITE set of registers, e.g. $\{A_i\}_{i=1}^{n}$, and a fixed FINITE program with instruction types

$a_i^{/} \longrightarrow$ NEXT S.I.

INCREMENT

a_i +:= 1

$a_j^{-} \nearrow$ NEXT INST

$a_j^{-} \twoheadrightarrow$ NEXT INSTR^N

DECREMENT & TEST

```
if a_j = 0   DO NEXT INSTR^N
  ELSE $( a_j  -:= 1
          DO NEXT INST
      $)
```

and HALT.

The program can be represented either with numerical instruction labels or as a planar graph. Execution starts with data values loaded into the registers at a nominated START instruction, then continues until HALT is executed.

A solution ctd)

The current configuration at any stage of a ~~[illegible]~~ register machine computation is represented by i) the program counter value

ii) the current register contents $\{a_i\}_{i=1}^{n}$.

The program takes as input a natural number s, and expresses it

EITHER a) $s = 0$ EXIT0 $a := 0$

OR b) $s = 2^{a'}(2s'+1)$ UNIQUELY,

take EXIT1, set $(a,s) := (a', s')$

This allows us to decode stack representations: BINARY representation of s: (assume $\neq 0$)

BASE OF STACK $\boxed{(\text{binary representation of } s')\ 1\ 0\ \ldots\ 0\ 0}$

a' zeroes

A solution) ctd)

Taking the same stack encoding revised to compute and decode $z = Z(x,y) = 2^x(2y+1) - 1$ gives a unique representation $[x,y]$ for all pairs. We may now represent both fixed data structures and lists of arbitrary length by unique nat'. nos.

A To code a program, number the nodes and choose a fixed representation for each instruction. Code the list that contains the node representations

B If the machine has n registers, code the list $(pc, a_1, a_2, \dots a_n)$. Usually $pc=1$ initially

---

A universal register machine has been presented in the notes that requires 3 input parameters; program code p, initial pc, and initial register contents

A solution) ctd)

That will immediately serve as the basis for the required machine, since the design handles exceptions such as wild branching and incomplete data specification.

What needs to be done is to specify the computation by the pair code [p, d], where p is coded so that the START instruction is #1. Then set $pc = 1$ and decode the pair to establish the other two arguments for the URM

The snag here is that candidates will lose time by giving much more detail than the marks available can justify. In a sense that's their problem, BUT . . .