Math 29: Computability Theory **Spring 2024**

PSET 1 — 04/05/2024

Prof. Miller ***Student: Amittai Siavava***

Problem 1.

The rational numbers, $\mathbb{Q}$, are $\left\{\frac{p}{q} \mid p, q \in \mathbb{Z}, q > 0\right\}$. The Gaussian rationals are complex nunbers of the form $\{r + si \mid r, s \in \mathbb{Q}\}$. Provide a bijection between the Gaussian rationals and $\mathbb{N}$.

Prove that it is a bijection.

We can provide a bijection between the Gaussian rationals and $\mathbb{N}$ by using the Cantor Pairing Function. The Cantor Pairing Function is defined as $p : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ with

$$p(x, y) = \frac{1}{2}(x + y)(x + y + 1) + y.$$

p is a bijection [a]. We can construct a correspondence between the Gaussian rationals to $\mathbb{N}$ as follows. First, write $r + si$ in the form $\frac{a}{b} + \frac{c}{d}i$ where $a, b, c, d \in \mathbb{Z}$ such that $b, d > 0$, a, b are coprime, and c, d are coprime. Since $r, s \in \mathbb{Q}$, this is possible. Define the map $\psi : \mathbb{Q}[i] \to \mathbb{N}$ as follows:

$$\begin{aligned} &\psi : \mathbb{Q}[i] \to \mathbb{N} \\ &\frac{a}{b} + \frac{c}{d}i \mapsto p\left(p(a, b), p(c, d)\right). \end{aligned}$$

To show ***bijectivity***, we must show that ψ is both ***injective*** and ***surjective***.

1. ***ψ is injective***:

   Suppose $\psi\left(\frac{a_1}{b_1} + \frac{c_1}{d_1}i\right) = \psi\left(\frac{a_2}{b_2} + \frac{c_2}{d_2}i\right)$.
   This means that $p\left(p(a_1, b_1), p(c_1, d_1)\right) = p\left(p(a_2, b_2), p(c_2, d_2)\right)$. However, the Cantor Pairing Function is a bijection, meaning that $p\left(p(a_1, b_1), p(c_1, d_1)\right) = p\left(p(a_2, b_2), p(c_2, d_2)\right)$ *if and only if* $p(a_1, b_1) = p(a_2, b_2)$ and $p(c_1, d_1) = p(c_2, d_2)$, which in turn implies that $a_1 = a_2$, $b_1 = b_2$, $c_1 = c_2$, and $d_1 = d_2$. Therefore,

   $$\psi\left(\frac{a_1}{b_1} + \frac{c_1}{d_1}i\right) = \psi\left(\frac{a_2}{b_2} + \frac{c_2}{d_2}i\right) \implies \frac{a_1}{b_1} + \frac{c_1}{d_1}i = \frac{a_2}{b_2} + \frac{c_2}{d_2}i.$$

2. ***ψ is surjective***:

   Let $n \in \mathbb{N}$. Since the Cantor Pairing Function is surjective, $\exists\, x, y \in \mathbb{N}$ such that $n = p(x, y)$. Likewise, $\exists\, a, b, c, d \in \mathbb{N}$ such that $x = p(a, b)$ and $y = p(c, d)$. Therefore, there exists some $\gamma := \frac{a}{b} + \frac{c}{d}i$ in $\mathbb{Q}[i]$ such that

   $$\psi(\gamma) = \psi\left(\frac{a}{b} + \frac{c}{d}i\right) = p(p(a, b), p(c, d)) = p(x, y) = n.$$

[a] We used this in class. Do I need to show that it is a bijection?

Problem 2.

Give a register machine which converges if there is a 2 in R_0 and diverges otherwise.

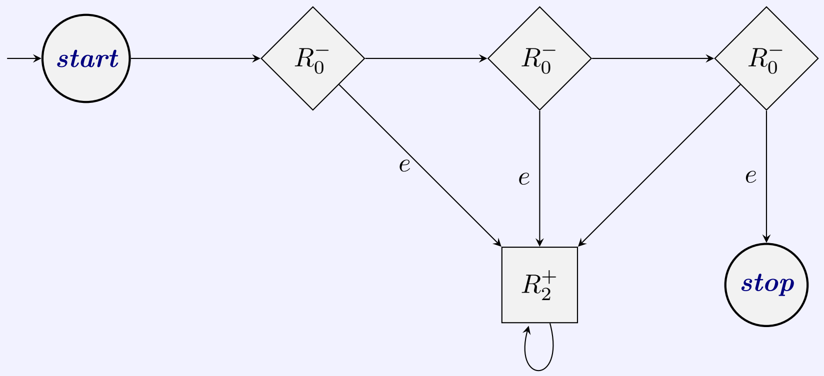

Figure 1: Converge if there is a 2 in R_0 and diverge otherwise.

Idea

The register machine in Figure 1 runs as follows:

1. Start at the ***start*** state. Let the initial value of R_0 be n.

2. Decrement R_0 by 1.

- If impossible, go to the looping state.
- If possible, the current value of R_0 is $n - 1$.

3. If step **2.** was possible, decrement R_0 by 1 again.

- If impossible, this means the value of R_0 before the decrement was $n - 1 = 0$, therefore $n = 1$, so $n \neq 2$. Go to the looping state.
- If possible, the current value of R_0 is $n - 2$.

4. If step **3.** was possible, decrement R_0 by 1 again.

- If impossible, this means the value of R_0 before the decrement was $n - 2 = 1$, therefore $n = 2$. Go to the ***stop*** state.
- If possible, the current value of R_0 is $n - 3 \geq 0$, meaning $n \geq 3 > 2$. Therefore, $n \neq 2$. Go to the looping state.

5. Finally, the looping node just increments R_2 *ad-infinitum*.

Problem 3.

Prove that the squaring function is computable by providing a register machine which takes in n in R_0 and outputs n^2 in R_1. You may use the multiplication function $x(n, m)$ — which starts with n in R_0 and m in R_1 and outputs $n \cdot m$ in R_2 — as a black box function labeled M.

Register Machine

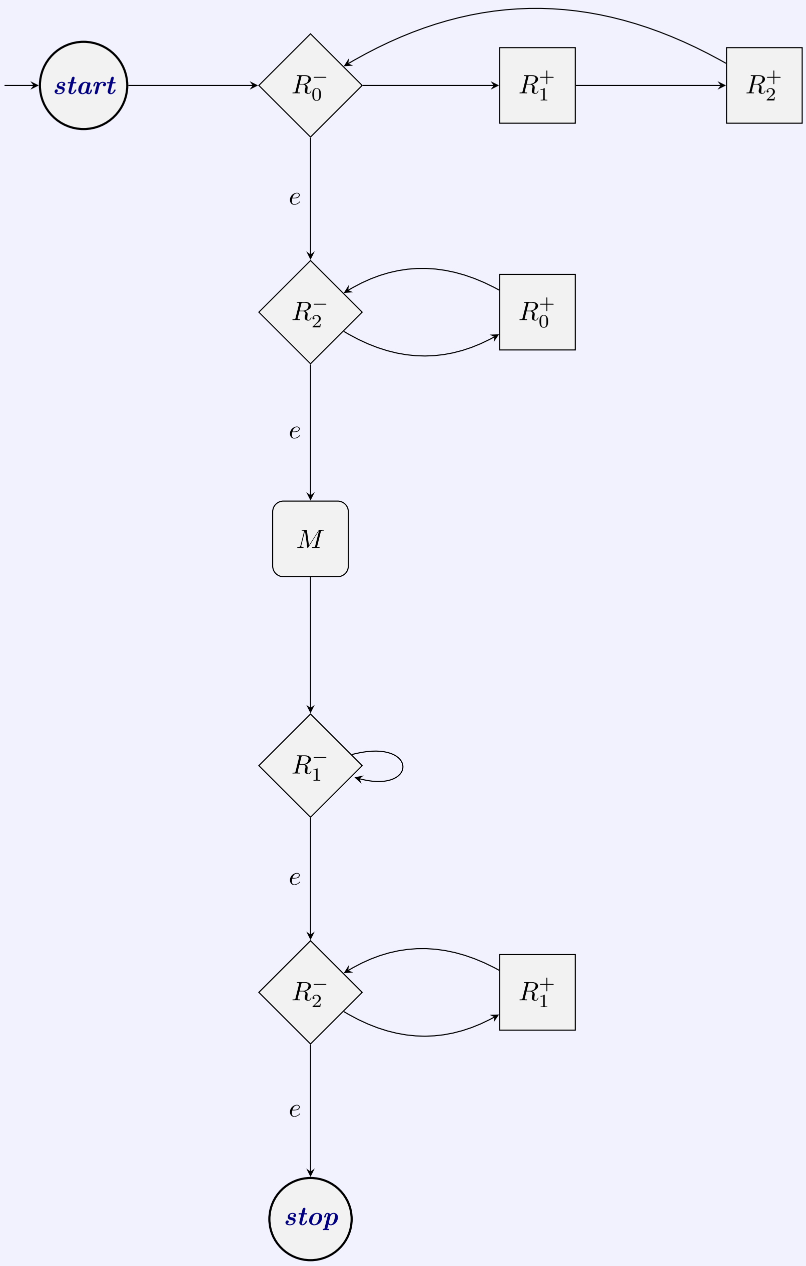

Figure 2: Calculate $f : n \mapsto n^2$ for $n \in \mathbb{N}$.

Idea

I found it useful to categorize the machine into five general parts (by height in Figure 2):

1. *start*
2. Copy n to R_1 and R_2, which sets $R_0 \leftarrow 0$.
3. Copy n from R_2 back to R_0.
4. Call M to calculate $R_2 \leftarrow R_0 \cdot R_1 = n \cdot n = n^2$.
5. Set $R_1 \leftarrow 0$.
6. Move n^2 from R_2 to R_1.
7. *stop*

Algorithm

I wrote this to convince myself that the machine works as I intended.

Please disregard if irrelevant.

Algorithm 1: Calculate $f : n \mapsto n^2$ for $n \in \mathbb{N}$.

start

while $(\textit{status} \coloneqq R_0 - 1) \neq e$ **do** ▷ `Copy` n `to` R_1 `and` R_2`, which sets` $R_0 = 0$

  $R_1 \leftarrow R_1 + 1$

  $R_2 \leftarrow R_2 + 1$

while $(\textit{status} \coloneqq R_2 - 1) \neq e$ **do** ▷ `Move` n `from` R_2 `back to` R_0

  $R_0 \leftarrow R_0 + 1$

call M ▷ `Call` M `to populate` R_2 `with` $n^2 \coloneqq R_0 \cdot R_1$

while $(\textit{status} \coloneqq R_1 - 1) \neq e$ **do** ▷ `Set` $R_1 \leftarrow 0$

  loop

while $(\textit{status} \coloneqq R_2 - 1) \neq e$ **do** ▷ `Move` n^2 `from` R_2 `to` R_1

  $R_1 \leftarrow R_1 + 1$

stop

Problem 4.

Prove that the set of ordered pairs of natural numbers (x, y) such that $x \leq y$ is computable by providing a register machine which takes n and m as imputs in R_0 and R_1 respectively and outputs 1 if $n \leq m$ and 0 in R_2 otherwise.

Register Machine

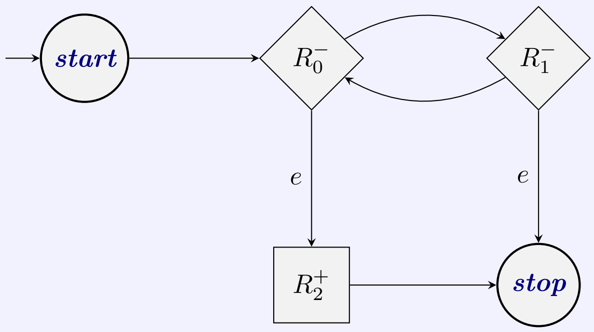

Figure 3: Calculate $f : (x, y) \mapsto \begin{cases} 1 & \text{if } x \leq y. \\ 0 & \text{otherwise.} \end{cases}$

Idea

The register machine in Figure 3 runs as follows:

1. ***start***
2. Deduct 1 from R_0
   - If impossible, $R_0 = 0$. Since we have deducted the same times from R_1 as we have from R_0 and R_1 has not run out before now, $R_1 \geq R_0$, meaning $R_0 \leq R_1$.
     Increment R_2 and go to the ***stop*** state.
   - If possible, go to step **3.**
3. Deduct 1 from R_1
   - If impossible, $R_1 = 0$. Since $R_0 \geq 0$ (as we have not branched out in the previous deduction from R_0 and gone to the ***stop*** state), $R_1 < R_0$, meaning $R_0 \nleq R_1$.
     Go to the ***stop*** state (leavng the default value of 0 in R_2).
   - If possible, go to step **2.**