Integer Gradient Descent

Nils Eckstein

Models of Computation

2020

Model Definition - Integer Gradient Descent

- Motivation: Gradient Descent is widely used in Machine Learning to optimize complex functions. Here we present one possible formalization of gradient descent as a model of computation.

Model Definition - Integer Gradient Descent

- Motivation: Gradient Descent is widely used in Machine Learning to optimize complex functions. Here we present one possible formalization of gradient descent as a model of computation.
- **Dynamics**:

$$x_{n+1} = x_n - \nabla F(x)\Big|_{x=x_n} \quad (1)$$

$$= x_n - \frac{d}{dx}F(x)\Big|_{x=x_n} \quad (2)$$

Model Definition - Integer Gradient Descent

- Motivation: Gradient Descent is widely used in Machine Learning to optimize complex functions. Here we present one possible formalization of gradient descent as a model of computation.
- **Dynamics**:

$$x_{n+1} = x_n - \nabla F(x)\Big|_{x=x_n} \tag{1}$$

$$= x_n - \frac{d}{dx}F(x)\Big|_{x=x_n} \tag{2}$$

- **The Program** is defined by giving:

$$\text{One differentiable function: } F(x) : \mathbb{R} \to \mathbb{R} \tag{3}$$

Model Definition - Integer Gradient Descent

- Motivation: Gradient Descent is widely used in Machine Learning to optimize complex functions. Here we present one possible formalization of gradient descent as a model of computation.
- **Dynamics**:

$$x_{n+1} = x_n - \nabla F(x)\Big|_{x=x_n} \quad (1)$$
$$= x_n - \frac{d}{dx}F(x)\Big|_{x=x_n} \quad (2)$$

- **The Program** is defined by giving:

$$\text{One differentiable function: } F(x) : \mathbb{R} \to \mathbb{R} \quad (3)$$

- **Input**:

$$\text{Initial value: } x_0 \in \mathbb{N}_0 \quad (4)$$
$$(5)$$

Model Definition - Integer Gradient Descent

- Motivation: Gradient Descent is widely used in Machine Learning to optimize complex functions. Here we present one possible formalization of gradient descent as a model of computation.
- **Dynamics**:

$$x_{n+1} = x_n - \nabla F(x)\Big|_{x=x_n} \tag{1}$$
$$= x_n - \frac{d}{dx}F(x)\Big|_{x=x_n} \tag{2}$$

- **The Program** is defined by giving:

$$\text{One differentiable function: } F(x) : \mathbb{R} \to \mathbb{R} \tag{3}$$

- **Input**:

$$\text{Initial value: } x_0 \in \mathbb{N}_0 \tag{4}$$
$$\tag{5}$$

- **Output**:

$$\text{An ordered list of integers: } \{x_n | x_n \in \mathbb{N}\} \tag{6}$$

Model Definition - Integer Gradient Descent

- Motivation: Gradient Descent is widely used in Machine Learning to optimize complex functions. Here we present one possible formalization of gradient descent as a model of computation.
- **Dynamics**:

$$x_{n+1} = x_n - \nabla F(x)\Big|_{x=x_n} \quad (1)$$

$$= x_n - \frac{d}{dx}F(x)\Big|_{x=x_n} \quad (2)$$

- **The Program** is defined by giving:

$$\text{One differentiable function: } F(x) : \mathbb{R} \to \mathbb{R} \quad (3)$$

- **Input**:

$$\text{Initial value: } x_0 \in \mathbb{N}_0 \quad (4)$$

$$(5)$$

- **Output**:

$$\text{An ordered list of integers: } \{x_n | x_n \in \mathbb{N}\} \quad (6)$$

- **HALTS**: If $x_{n+1} = x_n$

Model Definition - Integer Gradient Descent

Dynamics : $x_{n+1} = x_n - \frac{d}{dx} F(x_n)$

Program : $F : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

Model Definition - Integer Gradient Descent

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n)$

Program : $F : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Note that in contrast to the typical definition of gradient descent and without loss of generality we set the learning rate $\gamma = 1$ as we can incorporate constant factors into the definition of $F(x)$.

Model Definition - Integer Gradient Descent

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n)$
Program : $F : \mathbb{R} \to \mathbb{R}$, differentiable
Input : $x_0 \in \mathbb{N}$
Output : $x_n \in \mathbb{N}$
Halts : $x_m = x_{m+1}$

- Note that in contrast to the typical definition of gradient descent and without loss of generality we set the learning rate $\gamma = 1$ as we can incorporate constant factors into the definition of $F(x)$.
- Note also, that implicit in the definition is that **one program** is defined via **one function** for all inputs x_0, i.e. the function is not allowed to depend on input parameters. E.g. for a program that performs addition for two integers $a + b$, the function is not allowed to depend on a or b as we would have different functions for different inputs.

Model Definition - Integer Gradient Descent

Dynamics : $x_{n+1} = x_n - \frac{d}{dx} F(x_n)$

Program : $F : \mathbb{R} \rightarrow \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Note that in contrast to the typical definition of gradient descent and without loss of generality we set the learning rate $\gamma = 1$ as we can incorporate constant factors into the definition of $F(x)$.
- Note also, that implicit in the definition is that **one program** is defined via **one function** for all inputs x_0, i.e. the function is not allowed to depend on input parameters. E.g. for a program that performs addition for two integers $a + b$, the function is not allowed to depend on a or b as we would have different functions for different inputs.
- Finally, we can simplify the dynamics further by noting that we can choose $F(x) = (-\tilde{F}(x) + \frac{1}{2}x^2)$, leading to $x_{n+1} = x_n - \frac{d}{dx} F(x)\Big|_{x=x_n} = \frac{d}{dx}\tilde{F}(x)\Big|_{x=x_n}$. Leading to the simple **Dynamics**:

$$x_{n+1} = f(x_n) \tag{7}$$

$$\text{with } f(x_n) = \frac{d}{dx}\tilde{F}(x)\Big|_{x=x_n} \tag{8}$$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Define a function with:

$$\text{(EVEN)}\ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)}\ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1,\ \ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)}\ \frac{d}{dx}\tilde{F}(x) = x \text{ if x} = 1$$

$$\text{(ELSE)}\ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

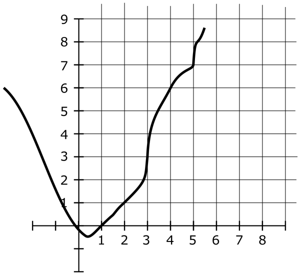
- Define a function with:

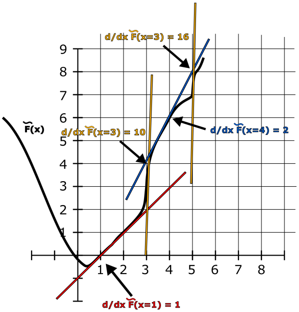
$$\text{(EVEN)} \quad \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)} \quad \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1,\ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)} \quad \frac{d}{dx}\tilde{F}(x) = x \text{ if } x = 1$$

$$\text{(ELSE)} \quad \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$





-

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Define a function with:

$$\text{(EVEN)} \ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)} \ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1, \ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)} \ \frac{d}{dx}\tilde{F}(x) = x \text{ if } x = 1$$

$$\text{(ELSE)} \ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$
Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable
Input : $x_0 \in \mathbb{N}$
Output : $x_n \in \mathbb{N}$
Halts : $x_m = x_{m+1}$

- Define a function with:

$$\text{(EVEN)}\ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)}\ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1,\ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)}\ \frac{d}{dx}\tilde{F}(x) = x \text{ if x} = 1$$

$$\text{(ELSE)}\ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

- With dynamics $x_{n+1} = \frac{d}{dx}\tilde{F}(x)$ and input $x_0 = 3$ we have:

Example: Collatz Sequence

$$\text{Dynamics}: x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$$
$$\text{Program}: F \text{ or } \tilde{F}: \mathbb{R} \to \mathbb{R}, \text{ differentiable}$$
$$\text{Input}: x_0 \in \mathbb{N}$$
$$\text{Output}: x_n \in \mathbb{N}$$
$$\text{Halts}: x_m = x_{m+1}$$

- Define a function with:

$$\text{(EVEN)}\ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$
$$\text{(UNEVEN)}\ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1,\ k \in \mathbb{N} \neq 0$$
$$\text{(ONE)}\ \frac{d}{dx}\tilde{F}(x) = x \text{ if } x = 1$$
$$\text{(ELSE)}\ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

- With dynamics $x_{n+1} = \frac{d}{dx}\tilde{F}(x)$ and input $x_0 = 3$ we have:
  - (x = 3, UNEVEN): $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Define a function with:

$$\text{(EVEN)}\ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)}\ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1,\ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)}\ \frac{d}{dx}\tilde{F}(x) = x \text{ if x} = 1$$

$$\text{(ELSE)}\ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

- With dynamics $x_{n+1} = \frac{d}{dx}\tilde{F}(x)$ and input $x_0 = 3$ we have:
  - (x = 3, UNEVEN): $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$
  - (x = 10, EVEN): $x_2 = x_1/2 = 10/2 = 5$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Define a function with:

$$\text{(EVEN)}\ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)}\ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1,\ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)}\ \frac{d}{dx}\tilde{F}(x) = x \text{ if } \mathrm{x} = 1$$

$$\text{(ELSE)}\ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

- With dynamics $x_{n+1} = \frac{d}{dx}\tilde{F}(x)$ and input $x_0 = 3$ we have:
  - (x = 3, UNEVEN): $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$
  - (x = 10, EVEN): $x_2 = x_1/2 = 10/2 = 5$
  - (x = 5, UNEVEN): $x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16$

Example: Collatz Sequence

$$\text{Dynamics}: x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$$
$$\text{Program}: F \text{ or } \tilde{F}: \mathbb{R} \to \mathbb{R}, \text{ differentiable}$$
$$\text{Input}: x_0 \in \mathbb{N}$$
$$\text{Output}: x_n \in \mathbb{N}$$
$$\text{Halts}: x_m = x_{m+1}$$

- Define a function with:

$$\text{(EVEN)} \ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$
$$\text{(UNEVEN)} \ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1, \ k \in \mathbb{N} \neq 0$$
$$\text{(ONE)} \ \frac{d}{dx}\tilde{F}(x) = x \text{ if } x = 1$$
$$\text{(ELSE)} \ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

- With dynamics $x_{n+1} = \frac{d}{dx}\tilde{F}(x)$ and input $x_0 = 3$ we have:
  - (x = 3, UNEVEN): $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$
  - (x = 10, EVEN): $x_2 = x_1/2 = 10/2 = 5$
  - (x = 5, UNEVEN): $x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16$
  - (x = 16, EVEN): $x_4 = x_3/2 = 16/2 = 8$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Define a function with:

$$\text{(EVEN)}\ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)}\ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1,\ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)}\ \frac{d}{dx}\tilde{F}(x) = x \text{ if x} = 1$$

$$\text{(ELSE)}\ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

- With dynamics $x_{n+1} = \frac{d}{dx}\tilde{F}(x)$ and input $x_0 = 3$ we have:
  - (x = 3, UNEVEN): $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$
  - (x = 10, EVEN): $x_2 = x_1/2 = 10/2 = 5$
  - (x = 5, UNEVEN): $x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16$
  - (x = 16, EVEN): $x_4 = x_3/2 = 16/2 = 8$
  - (x = 8, EVEN): $x_5 = x_4/2 = 8/2 = 4$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Define a function with:

$$\text{(EVEN)}\ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)}\ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1,\ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)}\ \frac{d}{dx}\tilde{F}(x) = x \text{ if } x = 1$$

$$\text{(ELSE)}\ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

- With dynamics $x_{n+1} = \frac{d}{dx}\tilde{F}(x)$ and input $x_0 = 3$ we have:
  - (x = 3, UNEVEN): $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$
  - (x = 10, EVEN): $x_2 = x_1/2 = 10/2 = 5$
  - (x = 5, UNEVEN): $x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16$
  - (x = 16, EVEN): $x_4 = x_3/2 = 16/2 = 8$
  - (x = 8, EVEN): $x_5 = x_4/2 = 8/2 = 4$
  - (x = 4, EVEN): $x_6 = x_5/2 = 4/2 = 2$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Define a function with:

$$\text{(EVEN)}\ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)}\ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k + 1,\ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)}\ \frac{d}{dx}\tilde{F}(x) = x \text{ if } \mathrm{x} = 1$$

$$\text{(ELSE)}\ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

- With dynamics $x_{n+1} = \frac{d}{dx}\tilde{F}(x)$ and input $x_0 = 3$ we have:
  - (x = 3, UNEVEN): $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$
  - (x = 10, EVEN): $x_2 = x_1/2 = 10/2 = 5$
  - (x = 5, UNEVEN): $x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16$
  - (x = 16, EVEN): $x_4 = x_3/2 = 16/2 = 8$
  - (x = 8, EVEN): $x_5 = x_4/2 = 8/2 = 4$
  - (x = 4, EVEN): $x_6 = x_5/2 = 4/2 = 2$
  - (x = 2, EVEN): $x_7 = x_6/2 = 2/2 = 1$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Define a function with:

$$\text{(EVEN)}\ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)}\ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k+1,\ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)}\ \frac{d}{dx}\tilde{F}(x) = x \text{ if } x = 1$$

$$\text{(ELSE)}\ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

- With dynamics $x_{n+1} = \frac{d}{dx}\tilde{F}(x)$ and input $x_0 = 3$ we have:
  - (x = 3, UNEVEN): $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$
  - (x = 10, EVEN): $x_2 = x_1/2 = 10/2 = 5$
  - (x = 5, UNEVEN): $x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16$
  - (x = 16, EVEN): $x_4 = x_3/2 = 16/2 = 8$
  - (x = 8, EVEN): $x_5 = x_4/2 = 8/2 = 4$
  - (x = 4, EVEN): $x_6 = x_5/2 = 4/2 = 2$
  - (x = 2, EVEN): $x_7 = x_6/2 = 2/2 = 1$
  - (x = 1, ONE): $x_8 = x_7$

Example: Collatz Sequence

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Define a function with:

$$\text{(EVEN)}\ \frac{d}{dx}\tilde{F}(x) = \frac{x}{2} \text{ for all } x = 2 \cdot k \in \mathbb{N}$$

$$\text{(UNEVEN)}\ \frac{d}{dx}\tilde{F}(x) = 3x + 1 \text{ for all } x = 2k+1,\ k \in \mathbb{N} \neq 0$$

$$\text{(ONE)}\ \frac{d}{dx}\tilde{F}(x) = x \text{ if } x = 1$$

$$\text{(ELSE)}\ \frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0 \text{ for all } x \in \mathbb{R} \setminus \mathbb{N}$$

- With dynamics $x_{n+1} = \frac{d}{dx}\tilde{F}(x)$ and input $x_0 = 3$ we have:
  - (x = 3, UNEVEN): $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$
  - (x = 10, EVEN): $x_2 = x_1/2 = 10/2 = 5$
  - (x = 5, UNEVEN): $x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16$
  - (x = 16, EVEN): $x_4 = x_3/2 = 16/2 = 8$
  - (x = 8, EVEN): $x_5 = x_4/2 = 8/2 = 4$
  - (x = 4, EVEN): $x_6 = x_5/2 = 4/2 = 2$
  - (x = 2, EVEN): $x_7 = x_6/2 = 2/2 = 1$
  - (x = 1, ONE): $x_8 = x_7$
  - (HALT)

Translating the Collatz Conjecture

Dynamics: $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Background:

Translating the Collatz Conjecture

Dynamics: $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Background:
  - **Banach Fixed Point Theorem:**
    Let (X, d) be a non-empty complete metric space with a contraction mapping $T : X \to X$. Then T admits a unique fixed point $x* \in X$ (i.e. $T(x*) = x*$). Furthermore $x*$ can be found as follows: start with an arbitrary element x_0 in X and define a sequence x_n by $x_n = T(x_n - 1)$ for $n \geq 1$. Then $x_n \to x*$. (Wikipedia)

Translating the Collatz Conjecture

Dynamics: $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Background:
  - **Banach Fixed Point Theorem:**
    Let (X, d) be a non-empty complete metric space with a contraction mapping $T : X \to X$. Then T admits a unique fixed point $x* \in X$ (i.e. $T(x*) = x*$). Furthermore $x*$ can be found as follows: start with an arbitrary element x_0 in X and define a sequence x_n by $x_n = T(x_n - 1)$ for $n \geq 1$. Then $x_n \to x*$. (Wikipedia)
    - **Contraction Mapping:**
      Let (X, d) be a complete metric space. Then a map $T : X \to X$ is called a contraction mapping on X if there exists $q \in [0, 1)$ such that $d(T(x), T(y)) \leq qd(x, y)$.

Translating the Collatz Conjecture

Dynamics: $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Background:
  - **Banach Fixed Point Theorem:**
    Let (X, d) be a non-empty complete metric space with a contraction mapping $T : X \to X$. Then T admits a unique fixed point $x* \in X$ (i.e. $T(x*) = x*$). Furthermore $x*$ can be found as follows: start with an arbitrary element x_0 in X and define a sequence x_n by $x_n = T(x_n - 1)$ for $n \geq 1$. Then $x_n \to x*$. (Wikipedia)
    - **Contraction Mapping:**
      Let (X, d) be a complete metric space. Then a map $T : X \to X$ is called a contraction mapping on X if there exists $q \in [0, 1)$ such that $d(T(x), T(y)) \leq qd(x, y)$.
- Thus, **if** we can find a metric d for which the function $T(x) = \frac{d}{dx}\tilde{F}(x)$, computing the Collatz sequence, is a contraction mapping, the Banach fixed point theorem says that repeated application of this function, will go to the unique fixed point of this function for all initial values $x_0 \in X$. By design the only fixed point this function has is x=1, thus proving the Collatz conjecture.

Translating the Collatz Conjecture

Dynamics: $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Background:
  - **Banach Fixed Point Theorem:**
    Let (X, d) be a non-empty complete metric space with a contraction mapping $T : X \to X$. Then T admits a unique fixed point $x* \in X$ (i.e. $T(x*) = x*$). Furthermore $x*$ can be found as follows: start with an arbitrary element x_0 in X and define a sequence x_n by $x_n = T(x_n - 1)$ for $n \geq 1$. Then $x_n \to x*$. (Wikipedia)
    - **Contraction Mapping:**
      Let (X, d) be a complete metric space. Then a map $T : X \to X$ is called a contraction mapping on X if there exists $q \in [0, 1)$ such that $d(T(x), T(y)) \leq qd(x, y)$.
- Thus, **if** we can find a metric d for which the function $T(x) = \frac{d}{dx}\tilde{F}(x)$, computing the Collatz sequence, is a contraction mapping, the Banach fixed point theorem says that repeated application of this function, will go to the unique fixed point of this function for all initial values $x_0 \in X$. By design the only fixed point this function has is x=1, thus proving the Collatz conjecture.
- However, $(\mathbb{R}, |\cdot|)$ does not trivially work as we would need $|\frac{d}{dx}\tilde{F}(x) - \frac{d}{dy}\tilde{F}(y)| \leq q|x - y|$

Translating the Collatz Conjecture

Dynamics: $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- Background:
  - **Banach Fixed Point Theorem:**
    Let (X, d) be a non-empty complete metric space with a contraction mapping $T : X \to X$. Then T admits a unique fixed point $x* \in X$ (i.e. $T(x*) = x*$). Furthermore $x*$ can be found as follows: start with an arbitrary element x_0 in X and define a sequence x_n by $x_n = T(x_n - 1)$ for $n \geq 1$. Then $x_n \to x*$. (Wikipedia)
    - **Contraction Mapping:**
      Let (X, d) be a complete metric space. Then a map $T : X \to X$ is called a contraction mapping on X if there exists $q \in [0, 1)$ such that $d(T(x), T(y)) \leq qd(x, y)$.
- Thus, **if** we can find a metric d for which the function $T(x) = \frac{d}{dx}\tilde{F}(x)$, computing the Collatz sequence, is a contraction mapping, the Banach fixed point theorem says that repeated application of this function, will go to the unique fixed point of this function for all initial values $x_0 \in X$. By design the only fixed point this function has is x=1, thus proving the Collatz conjecture.
- However, $(\mathbb{R}, |\cdot|)$ does not trivially work as we would need $|\frac{d}{dx}\tilde{F}(x) - \frac{d}{dy}\tilde{F}(y)| \leq q|x - y|$
- Instead: $|3x - 3y| > q|x - y|$ if x uneven and y uneven.

FRACTRANS - Motivation

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- The presented model is similar to FRACTRANS. A FRACTRAN program consists of an ordered list of fractions $\{f_1, f_2, ..., f_n\}$, $f_i \in \mathbb{Q}$ and the input $x_n \in \mathbb{N}$ is multiplied with these fractions. The first multiplication that yields an integer replaces x_n with $x_{n+1} = f_i \cdot x_n$.

FRACTRANS - Motivation

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \rightarrow \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- The presented model is similar to FRACTRANS. A FRACTRAN program consists of an ordered list of fractions $\{f_1, f_2, ..., f_n\}$, $f_i \in \mathbb{Q}$ and the input $x_n \in \mathbb{N}$ is multiplied with these fractions. The first multiplication that yields an integer replaces x_n with $x_{n+1} = f_i \cdot x_n$.
- Our model is similar in that we can interpret the FRACTRAN fractions $f_1, ..., f_n$ also as a set of linear functions $f^i(x) : \mathbb{R} \rightarrow \mathbb{R}$ that are evaluated in sequence: $x_{n+1} = f^i(x_n) = f_i \cdot x_n$ if $f^i(x_n) \in \mathbb{N}$. In our case the tangents of the function $\tilde{F}$: $\frac{d}{dx}\tilde{F}$ take the place of f^i.

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$
Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable
Input : $x_0 \in \mathbb{N}$
Output : $x_n \in \mathbb{N}$
Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx} F(x_n) = \frac{d}{dx} \tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:
    - (ADD) $\tilde{F}(x) = \frac{1}{3}x^2$ for all $x \in \mathbb{N}$.

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:
    - (ADD) $\tilde{F}(x) = \frac{1}{3}x^2$ for all $x \in \mathbb{N}$.
    - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = \frac{1}{3}2^k$, $k \in \mathbb{N}$.

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:
    - (ADD) $\tilde{F}(x) = \frac{1}{3}x^2$ for all $x \in \mathbb{N}$.
    - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = \frac{1}{3}2^k$, $k \in \mathbb{N}$.
    - (ELSE) $\frac{d}{dx}\tilde{F}(x)$ smooth everywhere else.

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:
    - (ADD) $\tilde{F}(x) = \frac{1}{3}x^2$ for all $x \in \mathbb{N}$.
    - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = \frac{1}{3}2^k$, $k \in \mathbb{N}$.
    - (ELSE) $\frac{d}{dx}\tilde{F}(x)$ smooth everywhere else.
  - Dynamics: $x_{n+1} = \frac{d}{dx}\tilde{F}(x_n) = \frac{2}{3}x_n$

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:
    - (ADD) $\tilde{F}(x) = \frac{1}{3}x^2$ for all $x \in \mathbb{N}$.
    - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = \frac{1}{3}2^k$, $k \in \mathbb{N}$.
    - (ELSE) $\frac{d}{dx}\tilde{F}(x)$ smooth everywhere else.
  - Dynamics: $x_{n+1} = \frac{d}{dx}\tilde{F}(x_n) = \frac{2}{3}x_n$
  - Input: $x_0 = 2^a 3^b$

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:
    - (ADD) $\tilde{F}(x) = \frac{1}{3}x^2$ for all $x \in \mathbb{N}$.
    - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = \frac{1}{3}2^k$, $k \in \mathbb{N}$.
    - (ELSE) $\frac{d}{dx}\tilde{F}(x)$ smooth everywhere else.
  - Dynamics: $x_{n+1} = \frac{d}{dx}\tilde{F}(x_n) = \frac{2}{3}x_n$
  - Input: $x_0 = 2^a 3^b$
    - (ADD) $x_1 = \frac{2}{3}x_0 = \frac{2}{3}2^a 3^b = 2^{a+1}3^{b-1}$

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:
    - (ADD) $\tilde{F}(x) = \frac{1}{3}x^2$ for all $x \in \mathbb{N}$.
    - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = \frac{1}{3}2^k$, $k \in \mathbb{N}$.
    - (ELSE) $\frac{d}{dx}\tilde{F}(x)$ smooth everywhere else.
  - Dynamics: $x_{n+1} = \frac{d}{dx}\tilde{F}(x_n) = \frac{2}{3}x_n$
  - Input: $x_0 = 2^a 3^b$
    - (ADD) $x_1 = \frac{2}{3}x_0 = \frac{2}{3}2^a 3^b = 2^{a+1}3^{b-1}$
    - (ADD) $x_2 = \frac{2}{3}x_1 = \frac{2}{3}2^{a+1}3^{b-1} = 2^{a+2}3^{b-2}$

Addition

$$\text{Dynamics}: x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$$
$$\text{Program}: F \text{ or } \tilde{F} : \mathbb{R} \to \mathbb{R}, \text{ differentiable}$$
$$\text{Input}: x_0 \in \mathbb{N}$$
$$\text{Output}: x_n \in \mathbb{N}$$
$$\text{Halts}: x_m = x_{m+1}$$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:
    - (ADD) $\tilde{F}(x) = \frac{1}{3}x^2$ for all $x \in \mathbb{N}$.
    - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = \frac{1}{3}2^k$, $k \in \mathbb{N}$.
    - (ELSE) $\frac{d}{dx}\tilde{F}(x)$ smooth everywhere else.
  - Dynamics: $x_{n+1} = \frac{d}{dx}\tilde{F}(x_n) = \frac{2}{3}x_n$
  - Input: $x_0 = 2^a 3^b$
    - (ADD) $x_1 = \frac{2}{3}x_0 = \frac{2}{3}2^a 3^b = 2^{a+1}3^{b-1}$
    - (ADD) $x_2 = \frac{2}{3}x_1 = \frac{2}{3}2^{a+1}3^{b-1} = 2^{a+2}3^{b-2}$
    - ...

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:
    - (ADD) $\tilde{F}(x) = \frac{1}{3}x^2$ for all $x \in \mathbb{N}$.
    - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = \frac{1}{3}2^k$, $k \in \mathbb{N}$.
    - (ELSE) $\frac{d}{dx}\tilde{F}(x)$ smooth everywhere else.
  - Dynamics: $x_{n+1} = \frac{d}{dx}\tilde{F}(x_n) = \frac{2}{3}x_n$
  - Input: $x_0 = 2^a 3^b$
    - (ADD) $x_1 = \frac{2}{3}x_0 = \frac{2}{3}2^a 3^b = 2^{a+1}3^{b-1}$
    - (ADD) $x_2 = \frac{2}{3}x_1 = \frac{2}{3}2^{a+1}3^{b-1} = 2^{a+2}3^{b-2}$
    - ...
    - (ADD) $x_b = \frac{2}{3}2^{a+b}3^0 = 2^{a+b+1}3^{-1} \notin \mathbb{N}$

Addition

Dynamics : $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$

Program : F or $\tilde{F} : \mathbb{R} \to \mathbb{R}$, differentiable

Input : $x_0 \in \mathbb{N}$

Output : $x_n \in \mathbb{N}$

Halts : $x_m = x_{m+1}$

- For **addition** the **FRACTRAN** program is $\{\frac{2}{3}\}$ for an input encoding of $x_0 = 2^a 3^b$ and will output 2^{a+b}.
- We can perform **addition** with **integer gradient descent** in the same way:
  - Program/Function:
    - (ADD) $\tilde{F}(x) = \frac{1}{3}x^2$ for all $x \in \mathbb{N}$.
    - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = \frac{1}{3}2^k$, $k \in \mathbb{N}$.
    - (ELSE) $\frac{d}{dx}\tilde{F}(x)$ smooth everywhere else.
  - Dynamics: $x_{n+1} = \frac{d}{dx}\tilde{F}(x_n) = \frac{2}{3}x_n$
  - Input: $x_0 = 2^a 3^b$
    - (ADD) $x_1 = \frac{2}{3}x_0 = \frac{2}{3}2^a 3^b = 2^{a+1}3^{b-1}$
    - (ADD) $x_2 = \frac{2}{3}x_1 = \frac{2}{3}2^{a+1}3^{b-1} = 2^{a+2}3^{b-2}$
    - ...
    - (ADD) $x_b = \frac{2}{3}2^{a+b}3^0 = 2^{a+b+1}3^{-1} \notin \mathbb{N}$
    - (HALT) $x_{b+1} = 2^{a+b+1}3^{-1} \notin \mathbb{N}$

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$

  - $i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$

  - $i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$

  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$

  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ $\quad (x_0)$
  - $\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ $\quad$ (INC3-DEC2)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)
  - $\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ (INC3-DEC2)
  - $\Rightarrow x_2 = \frac{3 \cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$ (INC3-DEC2)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)
  - $\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ (INC3-DEC2)
  - $\Rightarrow x_2 = \frac{3 \cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$ (INC3-DEC2)
  - $\Rightarrow x_3 = \frac{3 \cdot 11}{2} 2^1 3^2 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0$ (INC3-DEC2)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)
  - $\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ (INC3-DEC2)
  - $\Rightarrow x_2 = \frac{3 \cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$ (INC3-DEC2)
  - $\Rightarrow x_3 = \frac{3 \cdot 11}{2} 2^1 3^2 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0$ (INC3-DEC2)
  - $\Rightarrow x_4 = \frac{3 \cdot 11}{2} 2^0 3^3 7^2 11^3 13^0 = 2^{-1} 3^4 7^2 11^4 13^0$ (SWITCH1)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3\cdot 11}{2}x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2\cdot 13}{11\cdot 7\cdot 3}x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2\cdot 11}{3}x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2\cdot 7\cdot 11\cdot 13}x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)
  - $\Rightarrow x_1 = \frac{3\cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ (INC3-DEC2)
  - $\Rightarrow x_2 = \frac{3\cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$ (INC3-DEC2)
  - $\Rightarrow x_3 = \frac{3\cdot 11}{2} 2^1 3^2 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0$ (INC3-DEC2)
  - $\Rightarrow x_4 = \frac{3\cdot 11}{2} 2^0 3^3 7^2 11^3 13^0 = 2^{-1} 3^4 7^2 11^4 13^0$ (SWITCH1)
  - $\Rightarrow x_5 = \frac{2\cdot 13}{11\cdot 7\cdot 3} 2^{-1} 3^4 7^2 11^4 13^0 = 2^0 3^3 7^1 11^3 13^1$ (INC2-DEC3)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)
  - $\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ (INC3-DEC2)
  - $\Rightarrow x_2 = \frac{3 \cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$ (INC3-DEC2)
  - $\Rightarrow x_3 = \frac{3 \cdot 11}{2} 2^1 3^2 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0$ (INC3-DEC2)
  - $\Rightarrow x_4 = \frac{3 \cdot 11}{2} 2^0 3^3 7^2 11^3 13^0 = 2^{-1} 3^4 7^2 11^4 13^0$ (SWITCH1)
  - $\Rightarrow x_5 = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} 2^{-1} 3^4 7^2 11^4 13^0 = 2^0 3^3 7^1 11^3 13^1$ (INC2-DEC3)
  - $\Rightarrow x_6 = \frac{2 \cdot 11}{3} 2^0 3^3 7^1 11^3 13^1 = 2^1 3^2 7^1 11^4 13^1$ (INC2-DEC3)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)
  - $\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ (INC3-DEC2)
  - $\Rightarrow x_2 = \frac{3 \cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$ (INC3-DEC2)
  - $\Rightarrow x_3 = \frac{3 \cdot 11}{2} 2^1 3^2 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0$ (INC3-DEC2)
  - $\Rightarrow x_4 = \frac{3 \cdot 11}{2} 2^0 3^3 7^2 11^3 13^0 = 2^{-1} 3^4 7^2 11^4 13^0$ (SWITCH1)
  - $\Rightarrow x_5 = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} 2^{-1} 3^4 7^2 11^4 13^0 = 2^0 3^3 7^1 11^3 13^1$ (INC2-DEC3)
  - $\Rightarrow x_6 = \frac{2 \cdot 11}{3} 2^0 3^3 7^1 11^3 13^1 = 2^1 3^2 7^1 11^4 13^1$ (INC2-DEC3)
  - $\Rightarrow x_7 = \frac{2 \cdot 11}{3} 2^1 3^2 7^1 11^4 13^1 = 2^2 3^1 7^1 11^5 13^1$ (INC2-DEC3)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)
  - $\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ (INC3-DEC2)
  - $\Rightarrow x_2 = \frac{3 \cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$ (INC3-DEC2)
  - $\Rightarrow x_3 = \frac{3 \cdot 11}{2} 2^1 3^2 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0$ (INC3-DEC2)
  - $\Rightarrow x_4 = \frac{3 \cdot 11}{2} 2^0 3^3 7^2 11^3 13^0 = 2^{-1} 3^4 7^2 11^4 13^0$ (SWITCH1)
  - $\Rightarrow x_5 = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} 2^{-1} 3^4 7^2 11^4 13^0 = 2^0 3^3 7^1 11^3 13^1$ (INC2-DEC3)
  - $\Rightarrow x_6 = \frac{2 \cdot 11}{3} 2^0 3^3 7^1 11^3 13^1 = 2^1 3^2 7^1 11^4 13^1$ (INC2-DEC3)
  - $\Rightarrow x_7 = \frac{2 \cdot 11}{3} 2^1 3^2 7^1 11^4 13^1 = 2^2 3^1 7^1 11^5 13^1$ (INC2-DEC3)
  - $\Rightarrow x_8 = \frac{2 \cdot 11}{3} 2^2 3^1 7^1 11^5 13^1 = 2^3 3^0 7^1 11^6 13^1$ (INC2-DEC3)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)
  - $\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ (INC3-DEC2)
  - $\Rightarrow x_2 = \frac{3 \cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$ (INC3-DEC2)
  - $\Rightarrow x_3 = \frac{3 \cdot 11}{2} 2^1 3^2 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0$ (INC3-DEC2)
  - $\Rightarrow x_4 = \frac{3 \cdot 11}{2} 2^0 3^3 7^2 11^3 13^0 = 2^{-1} 3^4 7^2 11^4 13^0$ (SWITCH1)
  - $\Rightarrow x_5 = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} 2^{-1} 3^4 7^2 11^4 13^0 = 2^0 3^3 7^1 11^3 13^1$ (INC2-DEC3)
  - $\Rightarrow x_6 = \frac{2 \cdot 11}{3} 2^0 3^3 7^1 11^3 13^1 = 2^1 3^2 7^1 11^4 13^1$ (INC2-DEC3)
  - $\Rightarrow x_7 = \frac{2 \cdot 11}{3} 2^1 3^2 7^1 11^4 13^1 = 2^2 3^1 7^1 11^5 13^1$ (INC2-DEC3)
  - $\Rightarrow x_8 = \frac{2 \cdot 11}{3} 2^2 3^1 7^1 11^5 13^1 = 2^3 3^0 7^1 11^6 13^1$ (INC2-DEC3)
  - $\Rightarrow x_9 = \frac{2 \cdot 11}{3} 2^3 3^0 7^1 11^6 13^1 = 2^4 3^{-1} 7^1 11^7 13^1$ (SWITCH2)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)
  - $\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ (INC3-DEC2)
  - $\Rightarrow x_2 = \frac{3 \cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$ (INC3-DEC2)
  - $\Rightarrow x_3 = \frac{3 \cdot 11}{2} 2^1 3^2 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0$ (INC3-DEC2)
  - $\Rightarrow x_4 = \frac{3 \cdot 11}{2} 2^0 3^3 7^2 11^3 13^0 = 2^{-1} 3^4 7^2 11^4 13^0$ (SWITCH1)
  - $\Rightarrow x_5 = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} 2^{-1} 3^4 7^2 11^4 13^0 = 2^0 3^3 7^1 11^3 13^1$ (INC2-DEC3)
  - $\Rightarrow x_6 = \frac{2 \cdot 11}{3} 2^0 3^3 7^1 11^3 13^1 = 2^1 3^2 7^1 11^4 13^1$ (INC2-DEC3)
  - $\Rightarrow x_7 = \frac{2 \cdot 11}{3} 2^1 3^2 7^1 11^4 13^1 = 2^2 3^1 7^1 11^5 13^1$ (INC2-DEC3)
  - $\Rightarrow x_8 = \frac{2 \cdot 11}{3} 2^2 3^1 7^1 11^5 13^1 = 2^3 3^0 7^1 11^6 13^1$ (INC2-DEC3)
  - $\Rightarrow x_9 = \frac{2 \cdot 11}{3} 2^3 3^0 7^1 11^6 13^1 = 2^4 3^{-1} 7^1 11^7 13^1$ (SWITCH2)
  - $\Rightarrow x_{10} = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} 2^4 3^{-1} 7^1 11^7 13^1 = 2^3 3^0 7^0 11^6 13^0$ (HALT)

Looping and states: Multiplication

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input: $2^a 3^0 7^b 11^0 13^0$, $a \geq b$ the desired result will be stored in register 11: $11^{a \cdot b}$
- For that we need a function with:
  - (INC3-DEC2) $\frac{d}{dx}\tilde{F}(x) = \frac{3 \cdot 11}{2} x$ for all $x = 2^i 3^j 7^k 11^l 13^0$
  - (SWITCH1) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x$ for all $x = 2^{-1} 3^j 7^k 11^l 13^0$
  - (INC2-DEC3) $\frac{d}{dx}\tilde{F}(x) = \frac{2 \cdot 11}{3} x$ for all $x = 2^i 3^j 7^k 11^l 13^1$
  - (SWITCH2) $\frac{d}{dx}\tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x$ for all $x = 2^i 3^{-1} 7^k 11^l 13^1$
  - (HALT) $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = 2^i 3^j 7^0 11^l 13^m$
  - $i, j, l, m \in \mathbb{N}_0$, $k \in \mathbb{N}_{>0}$
- Example $3 \cdot 2$:
  - $x_0 = 2^3 3^0 7^2 11^0 13^0$ is of the form: A: $2^i 3^j 7^k 11^l 13^0$ (x_0)
  - $\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$ (INC3-DEC2)
  - $\Rightarrow x_2 = \frac{3 \cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$ (INC3-DEC2)
  - $\Rightarrow x_3 = \frac{3 \cdot 11}{2} 2^1 3^2 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0$ (INC3-DEC2)
  - $\Rightarrow x_4 = \frac{3 \cdot 11}{2} 2^0 3^3 7^2 11^3 13^0 = 2^{-1} 3^4 7^2 11^4 13^0$ (SWITCH1)
  - $\Rightarrow x_5 = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} 2^{-1} 3^4 7^2 11^4 13^0 = 2^0 3^3 7^1 11^3 13^1$ (INC2-DEC3)
  - $\Rightarrow x_6 = \frac{2 \cdot 11}{3} 2^0 3^3 7^1 11^3 13^1 = 2^1 3^2 7^1 11^4 13^1$ (INC2-DEC3)
  - $\Rightarrow x_7 = \frac{2 \cdot 11}{3} 2^1 3^2 7^1 11^4 13^1 = 2^2 3^1 7^1 11^5 13^1$ (INC2-DEC3)
  - $\Rightarrow x_8 = \frac{2 \cdot 11}{3} 2^2 3^1 7^1 11^5 13^1 = 2^3 3^0 7^1 11^6 13^1$ (INC2-DEC3)
  - $\Rightarrow x_9 = \frac{2 \cdot 11}{3} 2^3 3^0 7^1 11^6 13^1 = 2^4 3^{-1} 7^1 11^7 13^1$ (SWITCH2)
  - $\Rightarrow x_{10} = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} 2^4 3^{-1} 7^1 11^7 13^1 = 2^3 3^0 7^0 11^6 13^0$ (HALT)
  - $\Rightarrow x_{11} = 2^3 3^0 7^0 11^6 13^0 = x_{10}$ (HALT)

Turing Completeness

- Integer Gradient Descent is Turing complete as we can simulate every FRACTRAN program:

Turing Completeness

- Integer Gradient Descent is Turing complete as we can simulate every FRACTRAN program:
- Let $f = \{f_1, ..., f_n\}$ be an ordered list of fractions, a FRACTRAN program and initial value $n_0 = \prod_{k=0}^{N} p_k^{j_k} \in \mathbb{N}, p_k$ prime and $j_k \in \mathbb{N}_0$

Turing Completeness

- Integer Gradient Descent is Turing complete as we can simulate every FRACTRAN program:
- Let $f = \{f_1, ..., f_n\}$ be an ordered list of fractions, a FRACTRAN program and initial value $n_0 = \prod_{k=0}^{N} p_k^{j_k} \in \mathbb{N}, p_k$ prime and $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$ with $p_{fi} \notin \{p_0, ..., p_N\}$ prime.

Turing Completeness

- Integer Gradient Descent is Turing complete as we can simulate every FRACTRAN program:
- Let $f = \{f_1, ..., f_n\}$ be an ordered list of fractions, a FRACTRAN program and initial value $n_0 = \prod_{k=0}^{N} p_k^{j_k} \in \mathbb{N}, p_k$ prime and $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$ with $p_{fi} \notin \{p_0, ..., p_N\}$ prime.
- (START): $\frac{d}{dx}\tilde{F}(x) = f_1 x$ for all $x = k \cdot p_{f1}, k \in \mathbb{N}$

Turing Completeness

- Integer Gradient Descent is Turing complete as we can simulate every FRACTRAN program:
- Let $f = \{f_1, ..., f_n\}$ be an ordered list of fractions, a FRACTRAN program and initial value $n_0 = \prod_{k=0}^{N} p_k^{j_k} \in \mathbb{N}, p_k$ prime and $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$ with $p_{fi} \notin \{p_0, ..., p_N\}$ prime.
- (START): $\frac{d}{dx}\tilde{F}(x) = f_1 x$ for all $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f1 Q): $\frac{d}{dx}\tilde{F}(x) = f_1^{-1} f_2 \frac{p_{f2}}{p_{f1}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f1}, k \in \mathbb{N}$

Turing Completeness

- Integer Gradient Descent is Turing complete as we can simulate every FRACTRAN program:
- Let $f = \{f_1, ..., f_n\}$ be an ordered list of fractions, a FRACTRAN program and initial value $n_0 = \prod_{k=0}^{N} p_k^{j_k} \in \mathbb{N}, p_k$ prime and $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$ with $p_{fi} \notin \{p_0, ..., p_N\}$ prime.
- (START): $\frac{d}{dx}\tilde{F}(x) = f_1 x$ for all $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f1 Q): $\frac{d}{dx}\tilde{F}(x) = f_1^{-1} f_2 \frac{p_{f2}}{p_{f1}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f2 Q): $\frac{d}{dx}\tilde{F}(x) = f_2^{-1} f_3 \frac{p_{f3}}{p_{f2}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f2}, k \in \mathbb{N}$

Turing Completeness

- Integer Gradient Descent is Turing complete as we can simulate every FRACTRAN program:
- Let $f = \{f_1, ..., f_n\}$ be an ordered list of fractions, a FRACTRAN program and initial value $n_0 = \prod_{k=0}^{N} p_k^{j_k} \in \mathbb{N}, p_k$ prime and $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$ with $p_{fi} \notin \{p_0, ..., p_N\}$ prime.
- (START): $\frac{d}{dx}\tilde{F}(x) = f_1 x$ for all $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f1 Q): $\frac{d}{dx}\tilde{F}(x) = f_1^{-1} f_2 \frac{p_{f2}}{p_{f1}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f2 Q): $\frac{d}{dx}\tilde{F}(x) = f_2^{-1} f_3 \frac{p_{f3}}{p_{f2}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f2}, k \in \mathbb{N}$
- (fi Q): $\frac{d}{dx}\tilde{F}(x) = f_i^{-1} f_{i+1} \frac{p_{fi+1}}{p_{fi}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{fi}, k \in \mathbb{N}$

Turing Completeness

- Integer Gradient Descent is Turing complete as we can simulate every FRACTRAN program:
- Let $f = \{f_1, ..., f_n\}$ be an ordered list of fractions, a FRACTRAN program and initial value $n_0 = \prod_{k=0}^{N} p_k^{j_k} \in \mathbb{N}, p_k$ prime and $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$ with $p_{fi} \notin \{p_0, ..., p_N\}$ prime.
- (START): $\frac{d}{dx}\tilde{F}(x) = f_1 x$ for all $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f1 Q): $\frac{d}{dx}\tilde{F}(x) = f_1^{-1} f_2 \frac{p_{f2}}{p_{f1}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f2 Q): $\frac{d}{dx}\tilde{F}(x) = f_2^{-1} f_3 \frac{p_{f3}}{p_{f2}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f2}, k \in \mathbb{N}$
- (fi Q): $\frac{d}{dx}\tilde{F}(x) = f_i^{-1} f_{i+1} \frac{p_{fi+1}}{p_{fi}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{fi}, k \in \mathbb{N}$
- (fi N): $\frac{d}{dx}\tilde{F}(x) = \frac{p_{f1}}{p_{fi}} x$ for all $x \in \mathbb{N}$ and $x = k \cdot p_{fi}, p_{fi} \neq p_{f1}, k \in \mathbb{N}$

Turing Completeness

- Integer Gradient Descent is Turing complete as we can simulate every FRACTRAN program:
- Let $f = \{f_1, ..., f_n\}$ be an ordered list of fractions, a FRACTRAN program and initial value $n_0 = \prod_{k=0}^{N} p_k^{j_k} \in \mathbb{N}, p_k$ prime and $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$ with $p_{fi} \notin \{p_0, ..., p_N\}$ prime.
- (START): $\frac{d}{dx}\tilde{F}(x) = f_1 x$ for all $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f1 Q): $\frac{d}{dx}\tilde{F}(x) = f_1^{-1} f_2 \frac{p_{f2}}{p_{f1}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f2 Q): $\frac{d}{dx}\tilde{F}(x) = f_2^{-1} f_3 \frac{p_{f3}}{p_{f2}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f2}, k \in \mathbb{N}$
- (fi Q): $\frac{d}{dx}\tilde{F}(x) = f_i^{-1} f_{i+1} \frac{p_{fi+1}}{p_{fi}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{fi}, k \in \mathbb{N}$
- (fi N): $\frac{d}{dx}\tilde{F}(x) = \frac{p_{f1}}{p_{fi}} x$ for all $x \in \mathbb{N}$ and $x = k \cdot p_{fi}, p_{fi} \neq p_{f1}, k \in \mathbb{N}$
- (fn Q): $\frac{d}{dx}\tilde{F}(x) = \frac{p_{halt}}{p_{fn}} f_n^{-1} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{fn}, k \in \mathbb{N}$

Turing Completeness

- Integer Gradient Descent is Turing complete as we can simulate every FRACTRAN program:
- Let $f = \{f_1, ..., f_n\}$ be an ordered list of fractions, a FRACTRAN program and initial value $n_0 = \prod_{k=0}^{N} p_k^{j_k} \in \mathbb{N}, p_k$ prime and $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$ with $p_{fi} \notin \{p_0, ..., p_N\}$ prime.
- (START): $\frac{d}{dx}\tilde{F}(x) = f_1 x$ for all $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f1 Q): $\frac{d}{dx}\tilde{F}(x) = f_1^{-1} f_2 \frac{p_{f2}}{p_{f1}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f1}, k \in \mathbb{N}$
- (f2 Q): $\frac{d}{dx}\tilde{F}(x) = f_2^{-1} f_3 \frac{p_{f3}}{p_{f2}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{f2}, k \in \mathbb{N}$
- (fi Q): $\frac{d}{dx}\tilde{F}(x) = f_i^{-1} f_{i+1} \frac{p_{fi+1}}{p_{fi}} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{fi}, k \in \mathbb{N}$
- (fi N): $\frac{d}{dx}\tilde{F}(x) = \frac{p_{f1}}{p_{fi}} x$ for all $x \in \mathbb{N}$ and $x = k \cdot p_{fi}, p_{fi} \neq p_{f1}, k \in \mathbb{N}$
- (fn Q): $\frac{d}{dx}\tilde{F}(x) = \frac{p_{halt}}{p_{fn}} f_n^{-1} x$ for all $x \in \mathbb{Q} \setminus \mathbb{N}$ and $x = k \cdot p_{fn}, k \in \mathbb{N}$
- (HALT): $\frac{d}{dx}\tilde{F}(x) = x$ for all $x = k \cdot p_{halt}, k \in \mathbb{N}$