# Integer Gradient Descent

Nils Eckstein

Models of Computation

2020

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.

- 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:

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

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

- 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:

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

- 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:

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

Input:

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

(5)

- 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:

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

Input:

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

(5)

Output:

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

- 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:

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

Input:

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

(5)

Output:

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

▶ HALTS: If  $x_{n+1} = x_n$ 

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

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

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).

 $\begin{array}{ll} \operatorname{Dynamics}: x_{n+1} = x_n & -\frac{d}{d\epsilon} F(x_n) \\ \operatorname{Program}: F: \mathbb{R} \to \mathbb{R}, \text{ differentiable} \\ \operatorname{input}: x_0 \in \mathbb{N} \\ \operatorname{Output}: x_n \in \mathbb{N} \\ \operatorname{Halts}: x_m = x_{m+1} \end{array}$ 

- 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 x₀, 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.

- 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<sub>0</sub>, 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**:

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

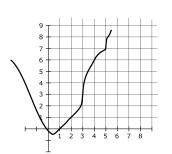
with 
$$f(x_n) = \frac{d}{dx} \tilde{F}(x) \Big|_{x=x_n}$$
 (8)

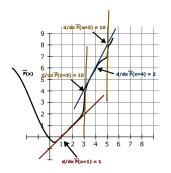
# Example: Collatz Sequence

 $\begin{array}{ll} \text{Dynamics}: x_{n+1} = x_{n} - \frac{d}{dx} f(x_{0}) = \frac{d}{dx} \tilde{F}(x_{0}) \\ \text{Program}: F \text{ or } \tilde{F}: \mathbb{R} \to \mathbb{R}, \text{ differentiable} \\ \text{Input}: x_{0} \in \mathbb{N} \\ \text{Output}: x_{0} \in \mathbb{N} \\ \text{Halts}: x_{m} = x_{m+1} \end{array}$ 

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

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





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

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

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

• (x = 3, UNEVEN): 
$$x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$$

(EVEN) 
$$\frac{d}{dx}\tilde{F}(x) = \frac{x}{2}$$
 for all  $x = 2 \cdot k \in \mathbb{N}$   
(UNEVEN)  $\frac{d}{dx}\tilde{F}(x) = 3x + 1$  for all  $x = 2k + 1, \ k \in \mathbb{N} \neq 0$   
(ONE)  $\frac{d}{dx}\tilde{F}(x) = x$  if  $x = 1$   
(ELSE)  $\frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0$  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):  $x_2 = x_1/2 = 10/2 = 5$

(EVEN) 
$$\frac{d}{dx}\tilde{F}(x) = \frac{x}{2}$$
 for all  $x = 2 \cdot k \in \mathbb{N}$   
(UNEVEN)  $\frac{d}{dx}\tilde{F}(x) = 3x + 1$  for all  $x = 2k + 1, \ k \in \mathbb{N} \neq 0$   
(ONE)  $\frac{d}{dx}\tilde{F}(x) = x$  if  $x = 1$   
(ELSE)  $\frac{d}{dx}\tilde{F}(x) \neq x \Leftrightarrow \frac{d}{dx}F(x) \neq 0$  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:
  - $\begin{array}{lll} (x=3, \, {\sf UNEVEN}): & x_1=3\cdot x_0+1=3\cdot 3+1=10 \\ (x=10, \, {\sf EVEN}): & x_2=x_1/2=10/2=5 \\ (x=5, \, {\sf UNEVEN}): & x_3=3\cdot x_2+1=3\cdot 5+1=16 \end{array}$

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

$$(x = 3, UNEVEN)$$
:  $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$ 

$$(x = 10, EVEN)$$
:  $x_2 = x_1/2 = 10/2 = 5$ 

(x = 16, EVEN): 
$$x_4 = x_3/2 = 16/2 = 8$$

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

$$(x = 3, UNEVEN)$$
:  $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$ 

$$(x = 10, EVEN)$$
:  $x_2 = x_1/2 = 10/2 = 5$ 

$$(x = 16, EVEN)$$
:  $x_4 = x_3/2 = 16/2 = 8$ 

$$(x = 8, EVEN)$$
:  $x_5 = x_4/2 = 8/2 = 4$ 

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

• (x = 3, UNEVEN): 
$$x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1$$

$$(x = 10, EVEN)$$
:  $x_2 = x_1/2 = 10/2 = 5$ 

$$\begin{array}{l} \text{ (x = 3, UNEVEN):} \\ \text{ (x = 10, EVEN):} \\ \text{ (x = 5, UNEVEN):} \\ \text{ (x = 5, UNEVEN):} \\ \text{ (x = 16, EVEN):} \\ \text{ (x = 16, EVEN):} \\ \text{ (x = 8, EVEN):} \\ \text{ (x = 4, EVEN):} \\ \end{array} \begin{array}{l} x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10 \\ x_2 = x_1/2 = 10/2 = 5 \\ x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16 \\ x_4 = x_3/2 = 16/2 = 8 \\ x_5 = x_4/2 = 8/2 = 4 \\ x_6 = x_5/2 = 4/2 = 2 \end{array}$$

$$(x = 16, EVEN)$$
:  $x_4 = x_3/2 = 16/2 = 8$ 

$$(x = 8, EVEN)$$
:  $x_5 = x_4/2 = 8/2 = 4$ 

$$(x = 4, EVEN)$$
:  $x_6 = x_5/2 = 4/2 = 2$ 

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

$$(x = 3, UNEVEN)$$
:  $x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10$ 

$$(x = 10, EVEN)$$
:  $x_2 = x_1/2 = 10/2 = 5$ 

$$\begin{array}{l} \text{ (x = 3, UNEVEN):} \\ \text{ (x = 10, EVEN):} \\ \text{ (x = 5, UNEVEN):} \\ \text{ (x = 5, UNEVEN):} \\ \text{ (x = 16, EVEN):} \\ \text{ (x = 16, EVEN):} \\ \text{ (x = 8, EVEN):} \\ \text{ (x = 4, EVEN):} \\ \text{ (x = 2, EVEN):} \\ \end{array} \begin{array}{l} x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10 \\ x_2 = x_1/2 = 10/2 = 5 \\ x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16 \\ x_4 = x_3/2 = 16/2 = 8 \\ x_5 = x_4/2 = 8/2 = 4 \\ x_6 = x_5/2 = 4/2 = 2 \\ x_7 = x_6/2 = 2/2 = 1 \end{array}$$

$$(x = 16, EVEN)$$
:  $x_4 = x_3/2 = 16/2 = 8$ 

$$(x = 8, EVEN)$$
:  $x_5 = x_4/2 = 8/2 = 4$   
 $(x = 4, EVEN)$ :  $x_6 = x_6/2 = 4/2 = 2$ 

$$(x = 4, EVEN)$$
:  $x_6 = x_5/2 = 4/2 = 2$ 

$$(x = 2, EVEN)$$
:  $x_7 = x_6/2 = 2/2 = 1$ 

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

• (x = 3, UNEVEN): 
$$x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 =$$

$$(x = 10, EVEN)$$
:  $x_2 = x_1/2 = 10/2 = 5$ 

$$\begin{array}{l} \text{ (x = 3, UNEVEN):} \\ \text{ (x = 10, EVEN):} \\ \text{ (x = 10, EVEN):} \\ \text{ (x = 5, UNEVEN):} \\ \text{ (x = 5, UNEVEN):} \\ \text{ (x = 16, EVEN):} \\ \text{ (x = 8, EVEN):} \\ \text{ (x = 4, EVEN):} \\ \text{ (x = 2, EVEN):} \\ \text{ (x = 1, ONE):} \\ \end{array} \begin{array}{l} x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10 \\ x_2 = x_1/2 = 10/2 = 5 \\ x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16 \\ x_4 = x_3/2 = 16/2 = 8 \\ x_4 = x_3/2 = 16/2 = 8 \\ x_5 = x_4/2 = 8/2 = 4 \\ x_6 = x_5/2 = 4/2 = 2 \\ x_7 = x_6/2 = 2/2 = 1 \\ x_8 = x_7 \end{array}$$

$$(x = 10, EVEN)$$
:  $x_4 = x_3/2 = 10/2 = 8$ 

$$(x = 4, EVEN)$$
:  $x_6 = x_5/2 = 4/2 = 2$ 

$$(x = 4, EVEN)$$
:  $x_6 = x_5/2 = 4/2 = 2$ 

$$(x = 2, EVEN):$$
  $x_7 = x_6/2 = 2/2 = 1$ 

$$(x = 1, ONE)$$
:  $x_8 = x_8$ 

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

$$\begin{array}{l} \text{ (x = 3, UNEVEN):} & x_1 = 3 \cdot x_0 + 1 = 3 \cdot 3 + 1 = 10 \\ \text{ (x = 10, EVEN):} & x_2 = x_1/2 = 10/2 = 5 \\ \text{ (x = 5, UNEVEN):} & x_3 = 3 \cdot x_2 + 1 = 3 \cdot 5 + 1 = 16 \\ \text{ (x = 16, EVEN):} & x_4 = x_3/2 = 16/2 = 8 \\ \text{ (x = 8, EVEN):} & x_5 = x_4/2 = 8/2 = 4 \\ \text{ (x = 4, EVEN):} & x_6 = x_5/2 = 4/2 = 2 \\ \text{ (x = 2, EVEN):} & x_7 = x_6/2 = 2/2 = 1 \\ \text{ (x = 1, ONE):} & x_8 = x_7 \\ \end{array}$$

# Translating the Collatz Conjecture

▶ Background:

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

# Translating the Collatz Conjecture

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

- 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)

#### 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  $n\ge 1$ . 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

 $\begin{array}{ll} \operatorname{Dynamicc:}_{n+1} = x_n - \frac{d}{d\epsilon} F(x_n) = \frac{d}{d\epsilon} \tilde{F}(x_n) \\ \operatorname{Pegram} F \circ \tilde{F} : \mathbb{R} \to \mathbb{R}, \text{ differentiable} \\ \operatorname{Input:}_{x_0} \in \mathbb{N} \\ \operatorname{Output:}_{x_0} \in \mathbb{N} \\ \operatorname{Halts:}_{x_0} : x_m = x_{m+1} \end{array}$ 

- 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  $d(T(x), T(y)) \le qd(x, y)$ .

▶ 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.

- 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  $n\ge 1$ . 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  $d(T(x), T(y)) \le qd(x, y)$ .

- ▶ 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  $|\frac{d}{dx}\tilde{F}(x) \frac{d}{dy}\tilde{F}(y)| \le q|x-y|$

- ► 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  $n\ge 1$ . 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 :  $x_{n+1} = x_0 - \frac{d}{dx}F(x_0) = \frac{d}{dx}F(x_0)$ Program :  $F \circ r F : \mathbb{R} \to \mathbb{R}$ , differentiable Input :  $x_0 \in \mathbb{N}$ Unique :  $x_0 \in \mathbb{N}$ 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

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

- ▶ 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  $f^i(x):\mathbb{R}\to\mathbb{R}$  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

youamics :  $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$ orgam: F or  $\tilde{F}: \mathbb{R} \to \mathbb{R}$ , differentiable put:  $x_0 \in \mathbb{N}$ when  $X_n \in \mathbb{N}$ when  $X_n \in \mathbb{N}$ 

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 :  $x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}F(x_n)$ Program : F or F :  $\mathbb{R} \to \mathbb{R}$ , differentiable nput :  $x_0 \in \mathbb{N}$ Jutput :  $x_0 \in \mathbb{N}$ Julput :  $x_0 = x_{m+1}$ 

- 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

Oynamics :  $x_{n+1} = x_0 - \frac{d}{dx}F(x_0) = \frac{d}{dx}F(x_0)$ Orogram : F or  $\hat{F}$  :  $\mathbb{R} \to \mathbb{R}$ , differentiable nout :  $x_0 \in \mathbb{N}$ Jutput :  $x_0 \in \mathbb{N}$ Jalts :  $x_m = x_{m+1}$ 

- For addition the FRACTRAN program is  $\{\frac{2}{3}\}$  for an input encoding of  $x_0=2^a3^b$  and will output  $2^{a+b}$ .
- ▶ We can perform addition with integer gradient descent in the same way:
  - Program/Function:

#### Addition

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

- For addition the FRACTRAN program is  $\{\frac{2}{3}\}$  for an input encoding of  $x_0=2^a3^b$  and will output  $2^{a+b}$ .
- ▶ We can perform addition with integer gradient descent in the same way:
  - Program/Function:
    - $\qquad \qquad \tilde{F}(x) = \tfrac{1}{3} x^2 \text{ for all } x \in \mathbb{N}.$

#### Addition

ynamics: 
$$x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$$
  
rogyam:  $F$  or  $\tilde{F}: \mathbb{R} \to \mathbb{R}$ , differentiable  
put:  $x_0 \in \mathbb{N}$   
utput:  $x_0 \in \mathbb{N}$   
alts:  $x_m = x_{m+1}$ 

- 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)  $\tilde{F}(x) = \frac{1}{3}x^2 \text{ for all } x \in \mathbb{N}.$   $\frac{d}{dx}\tilde{F}(x) = x \text{ for all } x = \frac{1}{3}2^k, k \in \mathbb{N}.$

#### Addition

ynamics: 
$$x_{n+1} = x_n - \frac{d}{dx}F(x_n) = \frac{d}{dx}\tilde{F}(x_n)$$
  
roopam:  $F$  or  $\tilde{F}: \mathbb{R} \to \mathbb{R}$ , differentiable  
put  $x_0 \in \mathbb{N}$   
utput:  $x_0 \in \mathbb{N}$   
alts:  $x_m = x_{m+1}$ 

- For addition the FRACTRAN program is  $\{\frac{2}{3}\}$  for an input encoding of  $x_0=2^a3^b$  and will output  $2^{a+b}$ .
- ▶ We can perform addition with integer gradient descent in the same way:
  - ► Program/Function:

$$\begin{array}{ll} \blacktriangleright & (\mathsf{ADD}) & \qquad \qquad \tilde{F}(x) = \frac{1}{3}x^2 \text{ for all } x \in \mathbb{N}. \\ \blacktriangleright & (\mathsf{HALT}) & \qquad \frac{d}{dt}\tilde{F}(x) = x \text{ for all } x = \frac{1}{3}2^k, \, k \in \mathbb{N}. \\ \blacktriangleright & (\mathsf{ELSE}) & \qquad \frac{d}{dt}\tilde{F}(x) \text{ smooth everywhere else.} \\ \end{array}$$

- ▶ 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:

$$\begin{array}{ll} \blacktriangleright \text{ (ADD)} & \qquad \qquad \tilde{F}(x) = \frac{1}{3}x^2 \text{ for all } x \in \mathbb{N}. \\ \blacktriangleright \text{ (HALT)} & \qquad \frac{d}{dx}\tilde{F}(x) = x \text{ for all } x = \frac{1}{3}2^k, \, k \in \mathbb{N}. \end{array}$$

(ELSE) 
$$\frac{dx}{dx}\tilde{F}(x)$$
 smooth everywhere else.

Dynamics: 
$$x_{n+1} = \frac{d}{dx}\tilde{F}(x_n) = \frac{2}{3}x_n$$

- ▶ 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:
    - $\begin{array}{ll} \qquad & \qquad \qquad \tilde{F}(x) = \frac{1}{3}x^2 \text{ for all } x \in \mathbb{N}. \\ \\ \blacktriangleright \text{ (HALT)} & \qquad \frac{d}{dx}\tilde{F}(x) = x \text{ for all } x = \frac{1}{3}2^k, k \in \mathbb{N}. \\ \\ \blacktriangleright \text{ (ELSE)} & \qquad \frac{d}{dx}\tilde{F}(x) \text{ smooth everywhere else.} \end{array}$
  - ▶ Dynamics:  $x_{n+1} = \frac{d}{dx}\tilde{F}(x_n) = \frac{2}{3}x_n$
  - Input:  $x_0 = 2^a 3^b$

- 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:

$$\begin{array}{ll} \tilde{F}(x) = \frac{1}{3}x^2 \text{ for all } x \in \mathbb{N}. \\ \\ \bullet \text{ (HALT)} & \frac{d}{dx}\tilde{F}(x) = x \text{ for all } x = \frac{1}{3}2^k, k \in \mathbb{N}. \\ \\ \bullet \text{ (ELSE)} & \frac{d}{dx}\tilde{F}(x) \text{ smooth everywhere else.} \end{array}$$

- ▶ Dynamics:  $x_{n+1} = \frac{d}{dx}\tilde{F}(x_n) = \frac{2}{3}x_n$
- Input:  $x_0 = 2^a 3^b$

$$(ADD) x_1 = \frac{2}{3}x_0 = \frac{2}{3}2^a 3^b = 2^{a+1}3^{b-1}$$

- 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:

$$\begin{array}{ll} \blacktriangleright & (\mathsf{ADD}) & \qquad \tilde{F}(x) = \frac{1}{3} \, x^2 \text{ for all } x \in \mathbb{N}. \\ \blacktriangleright & (\mathsf{HALT}) & \qquad \frac{d}{dx} \, \tilde{F}(x) = x \text{ for all } x = \frac{1}{3} \, 2^k, \, k \in \mathbb{N}. \\ \blacktriangleright & (\mathsf{ELSE}) & \qquad \frac{d}{dx} \, \tilde{F}(x) \text{ smooth everywhere else.} \end{array}$$

- **Dynamics**:  $x_{n+1} = \frac{d}{dx} \tilde{F}(x_n) = \frac{2}{3} x_n$
- ▶ Input:  $x_0 = 2^a 3^b$

• (ADD) 
$$x_1 = \frac{2}{3}x_0 = \frac{2}{3}2^a3^b = 2^{a+1}3^{b-1}$$

(ADD) 
$$x_2 = \frac{2}{3}x_1 = \frac{3}{2}2^{a+1}3^{b-1} = 2^{a+2}3^{b-2}$$

- ▶ 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:

$$\begin{array}{ll} & \text{(ADD)} & \qquad \tilde{F}(x) = \frac{1}{3}x^2 \text{ for all } x \in \mathbb{N}. \\ & \text{(HALT)} & \qquad \frac{d}{dx}\tilde{F}(x) = x \text{ for all } x = \frac{1}{3}2^k, k \in \mathbb{N}. \\ & \text{(ELSE)} & \qquad \frac{d}{dx}\tilde{F}(x) \text{ smooth everywhere else.} \end{array}$$

- Dynamics:  $x_{n+1} = \frac{d}{dx} \tilde{F}(x_n) = \frac{2}{3} x_n$
- Input:  $x_0 = 2^a 3^b$

• (ADD) 
$$x_1 = \frac{2}{3}x_0 = \frac{2}{3}2^a3^b = 2^{a+1}3^{b-1}$$

$$(ADD) x_2 = \frac{2}{3}x_1 = \frac{2}{3}2^{a+1}3^{b-1} = 2^{a+2}3^{b-2}$$

- 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) 
$$\tilde{F}(x) = \frac{1}{3}x^2 \text{ for all } x \in \mathbb{N}.$$

$$\text{(HALT)} \qquad \frac{d}{dx}\tilde{F}(x) = x \text{ for all } x = \frac{1}{3}2^k, k \in \mathbb{N}.$$

(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) 
$$x_1 = \frac{2}{3}x_0 = \frac{2}{3}2^a3^b = 2^{a+1}3^{b-1}$$

(ADD) 
$$x_2 = \frac{3}{3}x_1 = \frac{3}{3}2^{a+1}3^{b-1} = 2^{a+2}3^{b-2}$$

... 
$$(ADD) ... = 2a^{3} + b + 0 = 2a^{3} + b + 1 = 1 \neq N$$

... (ADD) 
$$x_b = \frac{2}{3}2^{a+b}3^0 = 2^{a+b+1}3^{-1} \notin \mathbb{N}$$

- 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) 
$$\tilde{F}(x) = \frac{1}{3}x^2 \text{ for all } x \in \mathbb{N}.$$

HALT) 
$$\frac{d}{dx}\tilde{F}(x) = x \text{ for all } x = \frac{1}{3}2^k, k \in \mathbb{N}.$$

$$(ELSE) \qquad \frac{d}{dx}\tilde{F}(x) \text{ smooth everywhere else.}$$

Dynamics: 
$$x_{n+1} = \frac{d}{dx} \tilde{F}(x_n) = \frac{2}{3} x_n$$

Input: 
$$x_0 = 2^a 3^b$$

$$ightharpoonup$$
 (ADD)  $x_1 = \frac{2}{2}x_0 = \frac{2}{2}2^a3^b = 2^{a+1}3^{b-1}$ 

(ADD) 
$$x_2 = \frac{2}{3}x_1 = \frac{3}{2}2^{a+1}3^{b-1} = 2^{a+2}3^{b-2}$$

... (ADD) 
$$x_b = \frac{2}{3}2^{a+b}3^0 = 2^{a+b+1}3^{-1} \notin \mathbb{N}$$

► (HALT) 
$$x_{b+1} = 2^{a+b+1}3^{-1} \notin \mathbb{N}$$

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

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

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- ▶ Input:  $2^a 3^0 7^b 11^0 13^0$ ,  $a \ge b$  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  $x = 2^{i}3^{j}7^{k}11^{l}13^{0}$ 

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

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input:  $2^a 3^0 7^b 11^0 13^0$ ,  $a \ge b$  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)  $\frac{d}{dx}\tilde{F}(x) = \frac{2\cdot 13}{11\cdot 17\cdot 3}x$  for all  $x = 2^{-1} 3^j 7^k 11^l 13^0$

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

- Similar to FRACTRANS (and thus register machines) we can store state in "registers", i.e. prime factors of the input to perform multiplication:
- Input:  $2^a 3^0 7^b 11^0 13^0$ ,  $a \ge b$  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  $x = 2^{-1}3^{j}7^{k}11^{j}13^{0}$
  - (INC2-DEC3)  $\frac{d}{dx}\tilde{F}(x) = \frac{2\cdot 11}{3}x$  for all  $x = 2^{i}3^{j}7^{k}11^{j}13^{1}$
  - $i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$

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

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

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

```
 \begin{array}{l} \blacktriangleright \text{ (INC3-DEC2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3\cdot 11}{2}x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ \blacktriangleright \text{ (SWITCH1)} & \frac{d}{dx} \tilde{F}(x) = \frac{2\cdot 13}{11\cdot 7\cdot 3}x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ \blacktriangleright \text{ (INC2-DEC3)} & \frac{d}{dx} \tilde{F}(x) = \frac{2\cdot 11}{3}x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ \blacktriangleright \text{ (SWITCH2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3}{2\cdot 7\cdot 11\cdot 13}x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ \blacktriangleright \text{ (HALT)} & \frac{d}{dx} \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ \blacktriangleright i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0} \end{array}
```

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

$$\begin{array}{ll} & \text{ (INC3-DEC2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3\cdot 11}{2}x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ & \text{ (SWITCH1)} & \frac{d}{dx} \tilde{F}(x) = \frac{2\cdot 13}{11\cdot 7\cdot 3}x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ & \text{ (INC2-DEC3)} & \frac{d}{dx} \tilde{F}(x) = \frac{2\cdot 11}{3}x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ & \text{ (SWITCH2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3}{2\cdot 7\cdot 11\cdot 13}x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ & \text{ (HALT)} & \frac{d}{dx} \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ & \text{ } i,j,l,m \in \mathbb{N}_0,k \in \mathbb{N}_{>0} \\ \end{array}$$

$$x_0 = 2^3 3^0 7^2 11^0 13^0$$
 is of the form: A:  $2^i 3^j 7^k 11^l 13^0$  (x<sub>0</sub>)

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

$$\begin{array}{l} \blacktriangleright \ \, (\mathsf{INC3\text{-}DEC2}) \qquad \qquad \frac{d}{dx} \, \tilde{F}(x) = \frac{3\cdot 11}{2} x \ \, \mathsf{for} \ \, \mathsf{all} \ \, x = 2^i 3^j 7^k 11^l 13^0 \\ \blacktriangleright \ \, (\mathsf{SWITCH1}) \qquad \qquad \frac{d}{dx} \, \tilde{F}(x) = \frac{2\cdot 13}{11\cdot 7\cdot 3} x \ \, \mathsf{for} \ \, \mathsf{all} \ \, x = 2^{-1} 3^j 7^k 11^l 13^0 \\ \blacktriangleright \ \, (\mathsf{INC2\text{-}DEC3}) \qquad \qquad \frac{d}{dx} \, \tilde{F}(x) = \frac{2\cdot 11}{3} x \ \, \mathsf{for} \ \, \mathsf{all} \ \, x = 2^i 3^j 7^k 11^l 13^1 \\ \blacktriangleright \ \, (\mathsf{SWITCH2}) \qquad \qquad \frac{d}{dx} \, \tilde{F}(x) = \frac{3}{2\cdot 7\cdot 11\cdot 13} x \ \, \mathsf{for} \ \, \mathsf{all} \ \, x = 2^i 3^{-1} 7^k 11^l 13^1 \\ \blacktriangleright \ \, (\mathsf{HALT}) \qquad \qquad \frac{d}{dx} \, \tilde{F}(x) = x \ \, \mathsf{for} \ \, \mathsf{all} \ \, x = 2^i 3^j 7^0 11^l 13^m \\ \blacktriangleright \ \, i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0} \end{array}$$

$$x_0 = 2^3 3^0 7^2 11^0 13^0$$
 is of the form: A:  $2^i 3^j 7^k 11^j 13^0$  (X<sub>0</sub>)  
 $x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$  (INC3-DEC2)

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

$$\begin{array}{ll} & \text{(INC3-DEC2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3 \cdot 11}{2} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ & \text{(SWITCH1)} & \frac{d}{dx} \tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ & \text{(INC2-DEC3)} & \frac{d}{dx} \tilde{F}(x) = \frac{2 \cdot 11}{3} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ & \text{(SWITCH2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ & \text{(HALT)} & \frac{d}{dx} \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ & \text{($i,j,l,m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$} \end{array}$$

$$\begin{array}{lll} & \chi_0 = 2^3 3^0 7^2 11^0 13^0 \text{ is of the form: A: } 2^i 3^j 7^k 11^l 13^0 \\ & \Rightarrow \chi_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0 \\ & \Rightarrow \chi_2 = \frac{3 \cdot 11}{2} 2^3 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0 \end{array} \qquad \begin{array}{ll} \text{(INC3-DEC2)} \\ & \Rightarrow \chi_2 = \frac{3 \cdot 11}{2} 2^3 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0 \end{array} \qquad \begin{array}{ll} \text{(INC3-DEC2)} \\ & \Rightarrow \chi_1 = \frac{3 \cdot 11}{2} 2^3 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0 \end{array} \qquad \begin{array}{ll} \text{(INC3-DEC2)} \\ & \Rightarrow \chi_2 = \frac{3 \cdot 11}{2} 2^3 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0 \end{array} \end{array}$$

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

$$\begin{array}{l} \blacktriangleright \text{ (INC3-DEC2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3\cdot 11}{2}x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ \blacktriangleright \text{ (SWITCH1)} & \frac{d}{dx} \tilde{F}(x) = \frac{2\cdot 13}{11\cdot 7\cdot 3}x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ \blacktriangleright \text{ (INC2-DEC3)} & \frac{d}{dx} \tilde{F}(x) = \frac{2\cdot 11}{3}x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ \blacktriangleright \text{ (SWITCH2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3}{2\cdot 7\cdot 11\cdot 13}x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ \blacktriangleright \text{ (HALT)} & \frac{d}{dx} \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ \blacktriangleright i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0} \end{array}$$

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

$$\begin{array}{ll} \bullet & (\mathsf{INC3\text{-}DEC2}) & \frac{d}{dx} \, \tilde{F}(x) = \frac{3\cdot 11}{2} x \; \mathsf{for} \; \mathsf{all} \; x = 2^i 3^j 7^k 11^l 13^0 \\ \bullet & (\mathsf{SWITCH1}) & \frac{d}{dx} \, \tilde{F}(x) = \frac{2\cdot 13}{11\cdot 73} x \; \mathsf{for} \; \mathsf{all} \; x = 2^{-1} 3^j 7^k 11^l 13^0 \\ \bullet & (\mathsf{INC2\text{-}DEC3}) & \frac{d}{dx} \, \tilde{F}(x) = \frac{2\cdot 11}{3} x \; \mathsf{for} \; \mathsf{all} \; x = 2^i 3^j 7^k 11^l 13^1 \\ \bullet & (\mathsf{SWITCH2}) & \frac{d}{dx} \, \tilde{F}(x) = \frac{3}{2\cdot 7\cdot 11\cdot 13} x \; \mathsf{for} \; \mathsf{all} \; x = 2^i 3^{-1} 7^k 11^l 13^1 \\ \bullet & (\mathsf{HALT}) & \frac{d}{dx} \, \tilde{F}(x) = x \; \mathsf{for} \; \mathsf{all} \; x = 2^i 3^j 7^0 11^l 13^m \\ \bullet & i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0} \end{array}$$

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

$$\begin{array}{ll} & \text{(INC3-DEC2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3 \cdot 11}{2} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ & \text{(SWITCH1)} & \frac{d}{dx} \tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ & \text{(INC2-DEC3)} & \frac{d}{dx} \tilde{F}(x) = \frac{2 \cdot 11}{3} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ & \text{(SWITCH2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ & \text{(HALT)} & \frac{d}{dx} \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ & \text{($i,j,l,m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$} \end{array}$$

Example 
$$3 \cdot 2$$
:

$$x_0 = 2^3 3^0 7^2 11^0 13^0 \text{ is of the form: A: } 2^i 3^j 7^k 11^l 13^0$$

$$\Rightarrow x_1 = \frac{3 \cdot 11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0$$

$$\Rightarrow x_2 = \frac{3 \cdot 11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0$$

$$\Rightarrow x_3 = \frac{3 \cdot 11}{2} 2^1 3^2 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0$$

$$\Rightarrow x_4 = \frac{3 \cdot 11}{2} 2^0 3^3 7^2 11^3 13^0 = 2^1 3^4 7^2 11^4 13^0$$

$$\Rightarrow x_5 = \frac{2 \cdot 13}{2} 2^0 3^3 7^2 11^4 13^0 = 2^0 3^3 7^1 11^3 13^1$$

$$(SWITCH1)$$

$$\Rightarrow x_5 = \frac{2 \cdot 13}{2} 2^{-1} 3^4 7^2 11^4 13^0 = 2^0 3^3 7^1 11^3 13^1$$

$$(INC2-DEC3)$$

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

$$\begin{array}{ll} & \text{(INC3-DEC2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3 \cdot 11}{2} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ & \text{(SWITCH1)} & \frac{d}{dx} \tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ & \text{(INC2-DEC3)} & \frac{d}{dx} \tilde{F}(x) = \frac{2 \cdot 11}{3} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ & \text{(SWITCH2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ & \text{(HALT)} & \frac{d}{dx} \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ & \text{($i,j,l,m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$} \end{array}$$

Example 
$$3 \cdot 2$$
:

$$\begin{array}{llll} & \chi_0 = 2^3 3^0 7^2 11^0 13^0 \text{ is of the form: A: } 2^i 3^j 7^k 11^l 13^0 & (\chi_0) \\ & \Rightarrow \chi_1 = \frac{3\cdot11}{2} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0 & (INC3-DEC2) \\ & \Rightarrow \chi_2 = \frac{3\cdot11}{2} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0 & (INC3-DEC2) \\ & \Rightarrow \chi_3 = \frac{3\cdot11}{2} 2^1 2^3 7^2 11^2 13^0 = 2^0 3^3 7^2 11^3 13^0 & (INC3-DEC2) \\ & \Rightarrow \chi_4 = \frac{3\cdot11}{2} 2^0 3^3 7^2 11^3 13^0 = 2^{-1} 3^4 7^2 11^4 13^0 & (SWITCH1) \\ & \Rightarrow \chi_5 = \frac{2\cdot11}{1\cdot1\cdot7\cdot3} 2^{-1} 3^4 7^2 11^4 13^0 = 2^0 3^3 7^1 11^3 13^1 & (INC2-DEC3) \\ & \Rightarrow \chi_6 = \frac{2\cdot11}{2} 2^0 3^3 7^1 11^3 13^1 = 2^1 3^2 7^1 11^4 13^1 & (INC2-DEC3) \end{array}$$

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

$$\begin{array}{ll} & \text{(INC3-DEC2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3 \cdot 11}{2} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ & \text{(SWITCH1)} & \frac{d}{dx} \tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ & \text{(INC2-DEC3)} & \frac{d}{dx} \tilde{F}(x) = \frac{2 \cdot 11}{3} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ & \text{(SWITCH2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ & \text{(HALT)} & \frac{d}{dx} \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ & \text{($i,j,l,m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$} \end{array}$$

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

$$\begin{array}{ll} & \text{(INC3-DEC2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3 \cdot 11}{2} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ & \text{(SWITCH1)} & \frac{d}{dx} \tilde{F}(x) = \frac{2 \cdot 13}{11 \cdot 7 \cdot 3} x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ & \text{(INC2-DEC3)} & \frac{d}{dx} \tilde{F}(x) = \frac{2 \cdot 11}{3} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ & \text{(SWITCH2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3}{2 \cdot 7 \cdot 11 \cdot 13} x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ & \text{(HALT)} & \frac{d}{dx} \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ & \text{($i,j,l,m \in \mathbb{N}_0, k \in \mathbb{N}_{>0}$} \end{array}$$

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

$$\begin{array}{l} \bullet \quad \text{(INC3-DEC2)} \qquad \frac{d}{dx} \, \tilde{F}(x) = \frac{3\cdot 11}{2} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ \bullet \quad \text{(SWITCH1)} \qquad \frac{d}{dx} \, \tilde{F}(x) = \frac{2\cdot 13}{11\cdot 7\cdot 3} x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ \bullet \quad \text{(INC2-DEC3)} \qquad \frac{d}{dx} \, \tilde{F}(x) = \frac{2\cdot 11}{3} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ \bullet \quad \text{(SWITCH2)} \qquad \frac{d}{dx} \, \tilde{F}(x) = \frac{3}{2\cdot 7\cdot 11\cdot 13} x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ \bullet \quad \text{(HALT)} \qquad \frac{d}{dx} \, \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ \bullet \quad i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0} \end{array}$$

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

$$\begin{array}{l} \blacktriangleright \text{ (INC3-DEC2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3\cdot 11}{2}x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ \blacktriangleright \text{ (SWITCH1)} & \frac{d}{dx} \tilde{F}(x) = \frac{2\cdot 13}{11\cdot 7\cdot 3}x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ \blacktriangleright \text{ (INC2-DEC3)} & \frac{d}{dx} \tilde{F}(x) = \frac{2\cdot 11}{3}x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ \blacktriangleright \text{ (SWITCH2)} & \frac{d}{dx} \tilde{F}(x) = \frac{3}{2\cdot 7\cdot 11\cdot 13}x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ \blacktriangleright \text{ (HALT)} & \frac{d}{dx} \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ \blacktriangleright i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0} \end{array}$$

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

$$\begin{array}{l} \bullet \quad \text{(INC3-DEC2)} \qquad \frac{d}{dx} \tilde{F}(x) = \frac{3\cdot 11}{2} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^0 \\ \bullet \quad \text{(SWITCH1)} \qquad \frac{d}{dx} \tilde{F}(x) = \frac{2\cdot 13}{11\cdot 7\cdot 3} x \text{ for all } x = 2^{-1} 3^j 7^k 11^l 13^0 \\ \bullet \quad \text{(INC2-DEC3)} \qquad \frac{d}{dx} \tilde{F}(x) = \frac{2\cdot 11}{3} x \text{ for all } x = 2^i 3^j 7^k 11^l 13^1 \\ \bullet \quad \text{(SWITCH2)} \qquad \frac{d}{dx} \tilde{F}(x) = \frac{3}{2\cdot 7\cdot 11\cdot 13} x \text{ for all } x = 2^i 3^{-1} 7^k 11^l 13^1 \\ \bullet \quad \text{(HALT)} \qquad \frac{d}{dx} \tilde{F}(x) = x \text{ for all } x = 2^i 3^j 7^0 11^l 13^m \\ \bullet \quad i, j, l, m \in \mathbb{N}_0, k \in \mathbb{N}_{>0} \end{array}$$

Example 
$$3 \cdot 2$$
:

$$\begin{array}{llll} & \chi_0 = 2^3 3^0 7^2 11^0 13^0 \text{ is of the form: A: } 2^i 3^j 7^k 11^l 13^0 & (\chi_0) \\ & \Rightarrow \chi_1 = \frac{3 \cdot 11}{2^1} 2^3 3^0 7^2 11^0 13^0 = 2^2 3^1 7^2 11^1 13^0 & (INC3-DEC2) \\ & \Rightarrow \chi_2 = \frac{3 \cdot 11}{2^1} 2^2 3^1 7^2 11^1 13^0 = 2^1 3^2 7^2 11^2 13^0 & (INC3-DEC2) \\ & \Rightarrow \chi_3 = \frac{3 \cdot 11}{2^1} 2^3 3^7 2^1 11^3 13^0 = 2^0 3^3 7^2 11^3 13^0 & (INC3-DEC2) \\ & \Rightarrow \chi_4 = \frac{3 \cdot 11}{2^1} 2^3 3^7 2^1 11^3 13^0 = 2^{-1} 3^4 7^2 11^4 13^0 & (SWITCH1) \\ & \Rightarrow \chi_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 \chi_6 = \frac{2 \cdot 11}{2^0} 2^3 3^7 11^3 13^1 = 2^1 3^2 7^1 11^4 13^1 & (INC2-DEC3) \\ & \Rightarrow \chi_7 = \frac{2 \cdot 11}{2^3} 2^3 2^7 11^4 13^1 = 2^3 3^7 11^5 13^1 & (INC2-DEC3) \\ & \Rightarrow \chi_8 = \frac{2 \cdot 11}{2^3} 2^3 2^7 11^5 13^1 = 2^3 3^0 7^1 11^5 13^1 & (INC2-DEC3) \\ & \Rightarrow \chi_9 = \frac{2 \cdot 11}{2^3} 2^3 3^0 7^1 11^6 13^1 = 2^4 3^{-1} 7^1 17^1 13^1 & (SWITCH2) \\ & \Rightarrow \chi_{10} = \frac{3}{2^3} \frac{2^3}{11 \cdot 11} 2^3 2^4 3^{-1} 7^1 17^1 13^1 = 2^3 3^0 7^0 11^6 13^0 & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 13^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{12} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & \Rightarrow \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & & \chi_{11} = 2^3 3^0 7^0 11^6 3^0 = \chi_{10} & (HALT) \\ & & \chi_{11} = 2^3 3^0 \gamma^0 11^0 13^0 & (HALT) \\ & & \chi_{12} = 2^3 3^0 \gamma^0 11^0 13^0 & (HALT) \\ & & \chi_{13} = 2^3 \gamma^0 \gamma^0 11^0 13^0 & (HALT) \\ & & \chi_{12} = 2^3 \gamma^0 \gamma^0 11^0 13^0 & (HALT) \\ &$$

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

- 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  $j_k \in \mathbb{N}_0$

- 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  $j_k \in \mathbb{N}_0$
- $ightharpoonup x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$  with  $p_{fi} \not\in \{p_0, ..., p_N\}$  prime.

- 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  $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f_1}^1 ... p_{f_i}^0 ... p_{f_n}^0$  with  $p_{f_i} \notin \{p_0, ..., p_N\}$  prime.
- ► (START):  $\frac{d}{dx}\tilde{F}(x) = f_1x$  for all  $x = k \cdot p_{f1}, k \in \mathbb{N}$

- 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  $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f_1}^1 ... p_{f_i}^0 ... p_{f_n}^0$  with  $p_{f_i} \notin \{p_0, ..., p_N\}$  prime.
- ► (START):  $\frac{d}{dx}\tilde{F}(x) = f_1x$  for all  $x = k \cdot p_{f_1}, k \in \mathbb{N}$
- ▶ (f1 Q):  $\frac{d}{dx}\tilde{F}(x) = f_1^{-1}f_2\frac{p_{f2}}{p_{f1}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{f1}, k \in \mathbb{N}$

- 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  $j_k \in \mathbb{N}_0$
- $x_0 = n_0 \cdot p_{f_1}^1 ... p_{f_i}^0 ... p_{f_n}^0$  with  $p_{f_i} \notin \{p_0, ..., p_N\}$  prime.
- ► (START):  $\frac{d}{dx}\tilde{F}(x) = f_1x$  for all  $x = k \cdot p_{f_1}, k \in \mathbb{N}$
- ▶ (f1 Q):  $\frac{d}{dx}\tilde{F}(x) = f_1^{-1}f_2\frac{p_{f2}}{p_{f1}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{f1}, k \in \mathbb{N}$
- ► (f2 Q):  $\frac{d}{dx}\tilde{F}(x) = f_2^{-1}f_3\frac{p_{f3}}{p_{f2}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{f2}, k \in \mathbb{N}$

- 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  $n_0 = \prod_{k=0}^N \rho_k^{j_k} \in \mathbb{N}, p_k$  prime and  $j_k \in \mathbb{N}_0$
- $ightharpoonup x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$  with  $p_{fi} \notin \{p_0, ..., p_N\}$  prime.
- ► (START):  $\frac{d}{dx}\tilde{F}(x) = f_1x$  for all  $x = k \cdot p_{f1}, k \in \mathbb{N}$
- ► (f1 Q):  $\frac{d}{dx}\tilde{F}(x) = f_1^{-1}f_2\frac{p_{f2}}{p_{f1}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{f1}, k \in \mathbb{N}$
- $\qquad \qquad \frac{d}{dx}\tilde{F}(x) = f_2^{-1}f_3\frac{p_{f3}}{p_{f2}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{f2}, k \in \mathbb{N}$
- $\qquad \qquad \qquad \frac{d}{dx}\tilde{F}(x) = f_i^{-1}f_{i+1}\frac{p_{fi+1}}{p_{fi}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{fi}, k \in \mathbb{N}$

- 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  $j_k \in \mathbb{N}_0$
- $ightharpoonup x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$  with  $p_{fi} \not \in \{p_0, ..., p_N\}$  prime.
- ► (START):  $\frac{d}{dx}\tilde{F}(x) = f_1x$  for all  $x = k \cdot p_{f_1}, k \in \mathbb{N}$
- ▶ (f1 Q):  $\frac{d}{dx}\tilde{F}(x) = f_1^{-1}f_2\frac{p_{f2}}{p_{f1}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{f1}, k \in \mathbb{N}$
- $\qquad \qquad \frac{d}{dx}\tilde{F}(x) = f_2^{-1}f_3\frac{p_{f3}}{p_{f2}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{f2}, k \in \mathbb{N}$
- $\qquad \qquad \qquad \frac{d}{dx}\tilde{F}(x) = f_i^{-1}f_{i+1}\frac{p_{fi+1}}{p_{fi}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{fi}, k \in \mathbb{N}$
- $\qquad \qquad \qquad \frac{d}{dx}\tilde{F}(x) = \frac{p_{f1}}{p_{fi}}x \text{ for all } x \in \mathbb{N} \text{ and } x = k \cdot p_{fi}, p_{fi} \neq p_{f1}, k \in \mathbb{N}$

- 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  $j_k \in \mathbb{N}_0$
- $ightharpoonup x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$  with  $p_{fi} \not\in \{p_0, ..., p_N\}$  prime.
- ► (START):  $\frac{d}{dx}\tilde{F}(x) = f_1x$  for all  $x = k \cdot p_{f_1}, k \in \mathbb{N}$
- ▶ (f1 Q):  $\frac{d}{dx}\tilde{F}(x) = f_1^{-1}f_2\frac{p_{f2}}{p_{f1}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{f1}, k \in \mathbb{N}$
- ▶ (f2 Q):  $\frac{d}{dx}\tilde{F}(x) = f_2^{-1}f_3\frac{p_{f3}}{p_{f2}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{f2}, k \in \mathbb{N}$
- $\qquad \qquad \qquad \frac{d}{dx}\tilde{F}(x) = f_i^{-1}f_{i+1}\frac{p_{fi+1}}{p_{fi}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{fi}, k \in \mathbb{N}$
- $\qquad \qquad \qquad \frac{d}{dx}\tilde{F}(x) = \frac{p_{f1}}{p_{fi}}x \text{ for all } x \in \mathbb{N} \text{ and } x = k \cdot p_{fi}, p_{fi} \neq p_{f1}, k \in \mathbb{N}$
- ▶ (fn Q):  $\frac{d}{dx}\tilde{F}(x) = \frac{p_{halt}}{p_{fn}}f_n^{-1}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{fn}, k \in \mathbb{N}$

- 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  $j_k \in \mathbb{N}_0$
- $ightharpoonup x_0 = n_0 \cdot p_{f1}^1 ... p_{fi}^0 ... p_{fn}^0$  with  $p_{fi} \not\in \{p_0, ..., p_N\}$  prime.
- ► (START):  $\frac{d}{dx}\tilde{F}(x) = f_1x$  for all  $x = k \cdot p_{f_1}, k \in \mathbb{N}$
- ► (f2 Q):  $\frac{d}{dx}\tilde{F}(x) = f_2^{-1}f_3\frac{p_{f3}}{p_{f2}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{f2}, k \in \mathbb{N}$
- $\qquad \qquad \qquad \frac{d}{dx}\tilde{F}(x) = f_i^{-1}f_{i+1}\frac{p_{fi+1}}{p_{fi}}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{fi}, k \in \mathbb{N}$
- ▶ (fi N):  $\frac{d}{dx}\tilde{F}(x) = \frac{p_{f1}}{p_{fi}}x \text{ for all } x \in \mathbb{N} \text{ and } x = k \cdot p_{fi}, p_{fi} \neq p_{f1}, k \in \mathbb{N}$
- $\qquad \qquad \qquad \frac{d}{dx}\tilde{F}(x) = \frac{p_{halt}}{p_{fn}}f_n^{-1}x \text{ for all } x \in \mathbb{Q} \setminus \mathbb{N} \text{ and } x = k \cdot p_{fn}, k \in \mathbb{N}$
- ► (HALT):  $\frac{d}{dx}\tilde{F}(x) = x$  for all  $x = k \cdot p_{halt}, k \in \mathbb{N}$