

# Disco

A Functional Programming Language for Discrete Mathematics



Brent A. Yorgey, Hendrix College

TFPIE 2023, Boston

# Disco



- ▶ Functional teaching language
- ▶ Designed for use in a Discrete Mathematics course
- ▶ Birthplace: TFPIE 2016, Maryland, USA!
- ▶ Have used it in a Discrete Math class once, in Spring 2022.
- ▶ Plan to use it again starting next week.

## Disco goals

- ▶ Teach early CS students basic FP concepts
- ▶ Help students connect math and computation
- ▶ Enhance learning with an interactive platform
- ▶ Minimize notational & conceptual friction

## Friction?

$$f : \mathbb{N} \rightarrow \mathbb{Q}$$

$$f(2n) = 0$$

$$f(2n+1) = \begin{cases} n/2 & \text{if } n > 5, \\ 3n+7 & \text{otherwise} \end{cases}$$

# Friction?

$$f : \mathbb{N} \rightarrow \mathbb{Q}$$
$$f(2n) = 0$$
$$f(2n + 1) = \begin{cases} n/2 & \text{if } n > 5, \\ 3n + 7 & \text{otherwise} \end{cases}$$

```
f :: Int -> Rational
f x
| even x      = 0
| n > 5      = fromIntegral n / 2
| otherwise  = 3*n + 7
where
  n = x `div` 2
```

# Friction?

$$f : \mathbb{N} \rightarrow \mathbb{Q}$$

$$f(2n) = 0$$

$$f(2n+1) = \begin{cases} n/2 & \text{if } n > 5, \\ 3n+7 & \text{otherwise} \end{cases}$$

$$f : \mathbb{N} \rightarrow \mathbb{Q}$$

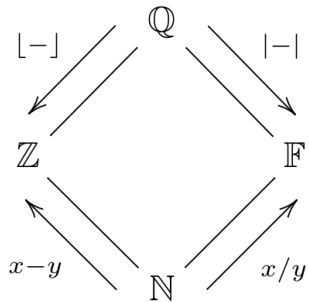
$$f(2n) = 0$$

$$f(2n+1) = \{? \begin{cases} n/2 & \text{if } n > 5, \\ 3n+7 & \text{otherwise} \end{cases} ?\}$$

Demo!

<https://replit.com/@BrentYorgey/Disco#README.md>

# Numeric types in Disco





## Issues / Future work

- ▶ Error messages!
- ▶ Showing multiple type instantiations
- ▶ Types vs sets

## Showing multiple type instantiations

What is the most general type of  $\lambda x.x - 2$ ?

## Showing multiple type instantiations

What is the most general type of  $\lambda x.x - 2$ ?

Internally, it is something like  $\forall a.(\text{HasSubtraction}(a), \mathbb{Z} <: a) \Rightarrow a \rightarrow a$

## Showing multiple type instantiations

What is the most general type of  $\lambda x.x - 2$ ?

Internally, it is something like  $\forall a. (\text{HasSubtraction}(a), \mathbb{Z} <: a) \Rightarrow a \rightarrow a$

... but we definitely don't want to show that to students!

## Showing multiple type instantiations

What is the most general type of  $\lambda x. x - 2$ ?

Internally, it is something like  $\forall a. (\text{HasSubtraction}(a), \mathbb{Z} <: a) \Rightarrow a \rightarrow a$

... but we definitely don't want to show that to students!

We do this instead, but it's confusing:

```
Disco> :type \x. x - 2
```

```
 $\lambda x. x - 2 : \mathbb{Z} \rightarrow \mathbb{Z}$ 
```

```
Disco> (\x. x - 2) (5/2)
```

```
1/2
```

## Showing multiple type instantiations

What about something like this instead?

```
Disco> :type \x. x - 2
```

```
 $\lambda x. x - 2$ 
```

```
:  $\mathbb{Z} \rightarrow \mathbb{Z}$ 
```

```
:  $\mathbb{Q} \rightarrow \mathbb{Q}$ 
```

## Types vs sets

- ▶  $\{2, 4, 7\}$  is an example of a \_\_\_\_\_

## Types vs sets

- ▶  $\{2, 4, 7\}$  is an example of a \_\_\_\_\_
- ▶  $\mathbb{N}$  is an example of a \_\_\_\_\_



## Types vs sets

- ▶  $\{2, 4, 7\}$  is an example of a \_\_\_\_\_
- ▶  $\mathbb{N}$  is an example of a \_\_\_\_\_
- ▶ In a math class,  $\mathbb{N} - \{2, 4, 7\}$  is a perfectly well-defined, countably infinite set.

## Types vs sets

- ▶  $\{2, 4, 7\}$  is an example of a \_\_\_\_\_
- ▶  $\mathbb{N}$  is an example of a \_\_\_\_\_
- ▶ In a math class,  $\mathbb{N} - \{2, 4, 7\}$  is a perfectly well-defined, countably infinite set.
- ▶ In Disco,  $\mathbb{N} - \{2, 4, 7\}$  is a syntax error!

## Types vs sets

- ▶  $\{2, 4, 7\}$  is an example of a \_\_\_\_\_
- ▶  $\mathbb{N}$  is an example of a \_\_\_\_\_
- ▶ In a math class,  $\mathbb{N} - \{2, 4, 7\}$  is a perfectly well-defined, countably infinite set.
- ▶ In Disco,  $\mathbb{N} - \{2, 4, 7\}$  is a syntax error!
- ▶ Why the difference, and how do we explain/frame it for students??

<https://github.com/disco-lang/disco>

