

# Functional Programming

## Functions

Prof. Dr. Peter Thiemann

Albert-Ludwigs-Universität Freiburg, Germany

SS 2019

# Function definition by cases

## Example: Absolute value

Find the absolute value of a number

- if  $x$  is positive, result is  $x$
- if  $x$  is negative, result is  $-x$

# Function definition by cases

## Example: Absolute value

Find the absolute value of a number

- if  $x$  is positive, result is  $x$
- if  $x$  is negative, result is  $-x$

## Definition

```
1  -- returns the absolute value of x
2  absolute :: Integer -> Integer
3  absolute x | x >= 0 = x
4  absolute x | x < 0 = - x
```

# Alternative styles of definition

## One equation

```
1 absolute' x | x >= 0 = x  
2           | x < 0 = -x
```

## Using if-then-else in an expression

```
1 absolute'' x = if x >= 0 then x else -x
```

# Recursion

Standard approach to define functions in functional languages (**no loops!**)

- Reduce a problem (e.g.,  $\text{power} \times n$ ) to a smaller problem of the same kind
- Eventually reach a base case that can be solved immediately
- Build up solutions from smaller solutions

## Example: power

Compute  $x^n$  without using the built-in operator

```
1 -- compute x to n-th power  
2 power x 0 = 1  
3 power x n | n > 0 = x * power x (n - 1)
```

## Example: Counting intersections

### Task

- Consider  $n$  non-parallel lines in the plane
- How often do these lines intersect (at most)? Call this number  $I(n)$ .

# Example: Counting intersections

## Task

- Consider  $n$  non-parallel lines in the plane
- How often do these lines intersect (at most)? Call this number  $I(n)$ .

Base case:  $n = 0$  (as simple as possible!)



# Example: Counting intersections

## Task

- Consider  $n$  non-parallel lines in the plane
- How often do these lines intersect (at most)? Call this number  $I(n)$ .

Base case:  $n = 0$  (as simple as possible!)

- Zero lines produce zero intersections:  $I(0) = 0$

# Example: Counting intersections

## Task

- Consider  $n$  non-parallel lines in the plane
- How often do these lines intersect (at most)? Call this number  $I(n)$ .

## Base case: $n = 0$ (as simple as possible!)

- Zero lines produce zero intersections:  $I(0) = 0$

## Inductive case: $n > 0$

# Example: Counting intersections

## Task

- Consider  $n$  non-parallel lines in the plane
- How often do these lines intersect (at most)? Call this number  $I(n)$ .

## Base case: $n = 0$ (as simple as possible!)

- Zero lines produce zero intersections:  $I(0) = 0$

## Inductive case: $n > 0$

- One line can intersect with the remaining lines at most  $n - 1$  times.

# Example: Counting intersections

## Task

- Consider  $n$  non-parallel lines in the plane
- How often do these lines intersect (at most)? Call this number  $I(n)$ .

## Base case: $n = 0$ (as simple as possible!)

- Zero lines produce zero intersections:  $I(0) = 0$

## Inductive case: $n > 0$

- One line can intersect with the remaining lines at most  $n - 1$  times.
- Remove this line. The remaining lines can intersect at most  $I(n - 1)$  times

# Example: Counting intersections

## Task

- Consider  $n$  non-parallel lines in the plane
- How often do these lines intersect (at most)? Call this number  $I(n)$ .

## Base case: $n = 0$ (as simple as possible!)

- Zero lines produce zero intersections:  $I(0) = 0$

## Inductive case: $n > 0$

- One line can intersect with the remaining lines at most  $n - 1$  times.
- Remove this line. The remaining lines can intersect at most  $I(n - 1)$  times
- Combine the above to  $I(n) = I(n - 1) + n - 1$

# Definition

## Counting intersections

```
1  -- max number of intersections of n lines
2  nisect :: Integer -> Integer
3  nisect 0 = 0
4  nisect n | n > 0 = nisect (n - 1) + n - 1
```

# Questions?

