Hand-in @ lecture #10

Jonas Smedegaard

2026-04-21

2.3.18a

For $f(x) = -3x + 4$ to be a bijection, it needs to be both one-to-one and onto.

Since $f(x)$ is a linear function with a negative gradient covering $\mathbb{R}$ for both domain and codomain, it is strictly decreasing. This means that it is indeed both one-to-one and onto, thus $f(x) = -3x + 4$ is a bijection from $\mathbb{R}$ to $\mathbb{R}$.

2.3.18b

For $f(x) = -3x^2 + 7$ to be a bijection, it needs to be both one-to-one and onto.

Since $f(x)$ is a second order polynomial with a negative second order coefficient, it is strictly increasing until a vertex and then strictly decreasing onwards. This means that it is not one-to-one since the codomain is 'folded' at the vertex meaning that except at the vertex it is two-to-one, nor is it onto since the domain is $\mathbb{R}$ but the codomain is a true subset of $\mathbb{R}$), thus $f(x) = -3x^2 + 7$ is not a bijection from $\mathbb{R}$ to $\mathbb{R}$.

A-3.2

> 'Prove Theorem 4, which states that for every nonzero real number x, the multiplicative inverse of x is unique.'

TODO

3.1.6

The following pseudocode describes an algorithm that takes as input a list of integers and finds the number of negative integers in the list:

```
integers := get_arg(0)
negatives := 0
foreach i in integers
  if i < 0
    negatives = negatives + 1
print negatives
```

3.1.24

The following pseudocode describes an algorithm that determines whether a function from a finite set to another finite set is one-to-one:

```
function := get_arg(0)
domain := get_arg(1)
results := new_array()
foreach i in domain
  res := execute_function(function, i)
  foreach item j in results
    if is_equal(res, j)
      print "function is not one-to-one"
      break
    append_to_array(results, res)
print "function is one-to-one"
```