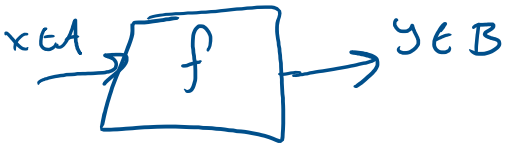
Lecture 2

January 11, 2024 10:55 AM

We want to study Models of Computation

* We have a function $f : A \to B$



* we want to "compute" f
    * is f "computable"?

* we have a pre-defined set of operations (that can be performed to "compute" f)

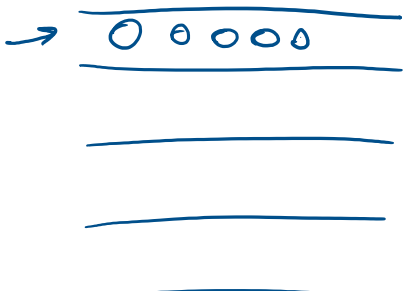
* These operations + the way we represent input/output define our computational model

* f is computable if there is a finite sequence of operations that $\forall x \in A$, the result of those operations would be $f(x)$

THE SCRAMBLE MACHINE

* The machine has an infinite number of rows and a lever

* the lever is initially pointing at the first row



* the machine has a single bit of memory (MEM = false, true)

Operations:

* LOWER_LEVER
* RAISE_LEVER
  - ↳ if already in top row, does nothing
* CHECK_EMPTY
  - ↳ checks whether current row is empty
  - ↳ if so, MEM = true, else MEM = false
* RESET
  - ↳ puts all the balls into the first row
* SCRAMBLE_DOWN
  - ↳ the balls above the lever start to fall to their next rows until no row (above the lever) has more balls than the one below it
  - ↳ also, MEM = True if any ball moved, false otherwise
* SCRAMBLE_UP
  - ↳ similar but reversed order

* RETURN_FALSE_IF_MEM_FALSE
* " " " TRUE
* " TRUE " " "
* " " " FALSE

The program terminates only if condition is satisfied

* LOOP
  - ↳ loops forever

Write a "program" that computes

$f: \mathbb{N} \to \{\text{true}, \text{false}\}$, $f(x) = \begin{cases} \text{true} & \text{if } x \text{ is even} \\ \text{false} & \text{if } x \text{ is odd} \end{cases}$

Solution:

LOWER_LEVEL
SCRAMBLE_DOWN
~~RAISE_LEVEL~~ X
SCRAMBLE_UP

RETURN_TRUE_IF_FALSE
RETURN_FALSE_IF_TRUE