Section 2

Encoding of Data Types into $\mathbb{N}$

- There are lots of different data types available.
- Some data types have finite size.
  - E.g. the type of Booleans $\{\mathrm{true}, \mathrm{false}\}$.
- Some data types have infinite size but are still "small".
  - E.g. the type of natural numbers $\mathbb{N} = \{0, 1, 2, \ldots, \}$.

Encoding of Data Types into $\mathbb{N}$

- Some data types are "big".
  - E.g. the set of subsets $\mathcal{P}(\mathbb{N})$ of $\mathbb{N}$.
  - Subsets of $\mathbb{N}$ have in general no finite description.
    - Some are finite (e.g. $\{0, 1, 3\}$).
    - Some can be described by formulae
      - E.g. the set of even numbers is

$$\{n \in \mathbb{N} \mid \exists m \in \mathbb{N}.n = 2m\} \enspace .$$

    - But there are subsets which cannot be described by formulae.
    - There is no way of associating a finite description to all elements of $\mathcal{P}(\mathbb{N})$.
      - This will be shown in this section.

Size and Computability

- We can introduce a notion of computability for finite and for small infinite data types.
  - E.g. it makes sense to compute certain functions mapping natural numbers to natural numbers.
- We cannot introduce in general a notion of computability for big data types.
  - We cannot even represent its elements on the computer.

Size and Computability

- There are notions of computability for certain "big data types" which make use of approximations of elements of such data types.
  - Topic of intensive research in Swansea esp. of Ulrich Berger, Jens Blanck, Monika Seisenberger, John Tucker.
  - One considers especially $\mathbb{R}$ and sets of functions (E.g. $\mathbb{N} \to \mathbb{N}$, $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$).
  - Not part of this lecture.

Topic of this Section

- In this Section we will make precise the notion of size of a set.
  - Notion of “cardinality” and “equinumerous”.
  - We will introduce a hierarchy of sizes.
    - We will be able to distinguish between sizes of different “big” sets.
- Countable sets will be the sets, which were called “small” above.
  - This notion will include the finite sets.

Structure of this Section

(a) Mathematical background.

(b) Cardinality.

(c) Countable sets.

(d) Reducing computability to $\mathbb{N}$.

(e) Encoding of some data types into $\mathbb{N}$.

(f) Further mathematical background: Partial functions.

Notions of Computability

- We will later introduce computability on $\mathbb{N}$.
- Computability on countable sets will in this section be reduced to computability on $\mathbb{N}$.

(a) Mathematical Background

Some Standard Sets

- $\mathbb{N}$ is the set of natural numbers:

$$\mathbb{N} := \{0, 1, 2, \ldots\} \ .$$

  - Note that 0 is a natural number.
  - When counting, we start with 0:
    - The element no. 0 of a sequence is what is usually called the first element:
      E.g., in $x_0, \ldots, x_{n-1}$, x_0 is the first variable.
    - The element no. 1 of a sequence is what is usually called the second element.
      E.g., in $x_0, \ldots, x_{n-1}$, x_1 is the second variable.
    - etc.

Some Standard Sets

- $\mathbb{Z}$ is the set of integers:

$$\mathbb{Z} := \mathbb{N} \cup \{-n \mid n \in \mathbb{N}\} \quad .$$

So

$$\mathbb{Z} = \{\ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \ldots, \}$$

Some Standard Sets

- $\mathbb{R}$ is the set of real numbers.
  - E.g.
    - $0.333333 \cdots \in \mathbb{R}$,
    - $\sqrt{2} \in \mathbb{R}$,
    - $-\sqrt{2} \in \mathbb{R}$,
    - $\pi \in \mathbb{R}$.

Some Standard Sets

- $\mathbb{Q}$ is the set of rationals, i.e.

$$\mathbb{Q} := \{\frac{x}{y} \mid x \in \mathbb{Z}, y \in \mathbb{N}, y \neq 0\} \quad .$$

  - So $\mathbb{Q}$ contains $\frac{2}{17}$, $\frac{-3}{5}$, $\frac{-2}{3}$, etc.
  - As usual we identify equal fractions e.g.

$$\frac{2}{4} = \frac{1}{2} \quad .$$

  - We write $-\frac{n}{m}$ instead of $\frac{-n}{m}$, e.g. $-\frac{1}{2} = \frac{-1}{2}$.
  - As usual $\frac{z}{-m} := -\frac{z}{m}$, e.g. $\frac{1}{-2} := -\frac{1}{2}$.

Some Standard Sets

- Assume A, B are sets.
  - $A \times B$ is the product of A and B:

$$A \times B := \{(x, y) \mid x \in A \wedge y \in B\}$$

  - $A \to B$ is the set of functions $f : A \to B$.

Some Standard Sets

- Assume A is a set, $k \in \mathbb{N}$.
  Then A^k is the **set of k-tuples of elements of A** or **k-fold Cartesian product of A** defined as follows:

  $$A^k := \{(x_0, \ldots, x_{k-1}) \mid x_0, \ldots, x_{k-1} \in A\} \enspace .$$

  Note that

  $$A^0 = \{()\}$$

  We identify A^1 with A.
  So we don't distinguish between (x) and x.

  Essentially, $A^k = \underbrace{A \times \cdots \times A}_{k \text{ times}}$.

A^*

Remark:

- A^* can be considered as the set of strings having letters in the alphabet A.
  - E.g. if

    $$A = \{a, b, c, \ldots, z\} \enspace ,$$

    then A^* is the set of strings formed from lower case letters.
  - So (r, e, d) stands for the string "red".
- A^k is the set of strings of length k from alphabet A.

A^*

- We define

  $$A^* := \{(a_0, \ldots, a_{k-1}) \mid k \in \mathbb{N}, a_0, \ldots, a_{k-1} \in A\}$$

  So A^* is
  - the **set of sequences of elements of A** (of arbitrary length),
  - also called the set of **lists of A**,
  - or **A-Kleene-Star**.
- So A^* is the union of all A^k for $k \in \mathbb{N}$, i.e.

  $$A^* = \bigcup_{k \in \mathbb{N}} A^k$$

$\mathcal{P}(X)$

- $\mathcal{P}(X)$, the powerset of X, is the set of all subsets of X.
- For finite sets X, the power set of X will be finite:

  $$\begin{aligned} \mathcal{P}(\{0,1,2\}) = \{ & \{\}, \\ & \{0\}, \{1\}, \{2\}, \\ & \{0,1\}, \{0,2\}, \{1,2\} \\ & \{0,1,2\}\} \end{aligned}$$

- For infinite sets X we will see that the X is big ("uncountable").
- Therefore we cannot write down the elements of $\mathcal{P}(X)$ for such X.

Exercise

- Write down $\mathcal{P}(\{0,1,2,3\})$ and $\mathcal{P}(\{0,1,2,3,4\})$.
- Make sure you have the right number of elements:
  If a set has n elements, then $\mathcal{P}(X)$ has 2^n elements.

Image of f

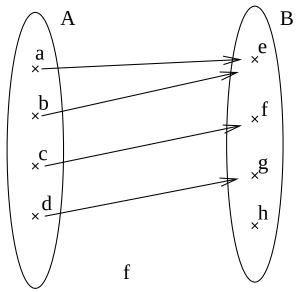


Image of f

Definition 2.1
Let $f : A \to B$, $C \subseteq A$.

(a) $f[C] := \{f(a) \mid a \in C\}$ *is called the* **image of C under f**.

(b) *The image of A under f (i.e. $f[A]$) is called the* **image of f**.

Image of f

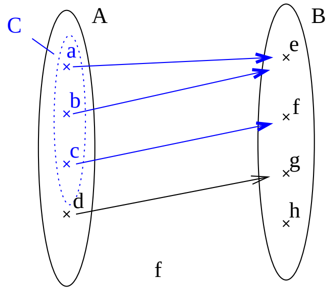


Image of C under f.

Image of f

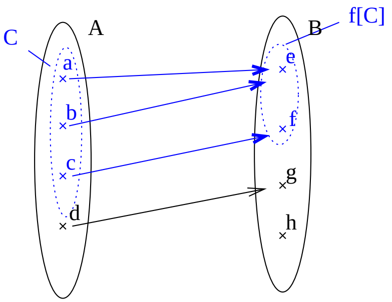


Image of C under f.

$f[C] = \{e, f\}$

Injective/Surjective/Bijective

Definition 2.2
Let A, B be sets, $f : A \to B$.

(a) f *is* **injective** *or* **an injection** *or* **one-to-one**, *if* f *applied to different elements of A has different results.* $\forall a, b \in A. a \neq b \to f(a) \neq f(b)$.

(b) f *is* **surjective** *or* **a surjection** *or* **onto**, *if every element of B is in the image of f:* $\forall b \in B. \exists a \in A. f(a) = b$.

(c) f *is* **bijective** *or* **a bijection** *or a* **one-to-one correspondence** *if it is both surjective and injective.*

Image of f

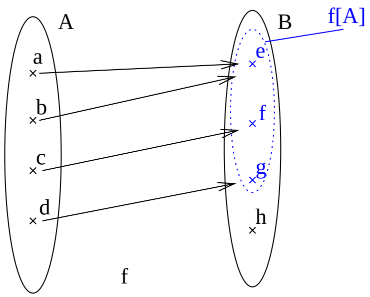


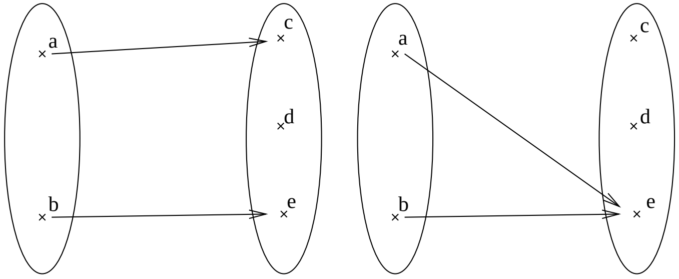
Image of f.

$f[A] = \{e, f, g\}$

Visualisation of "Injective"

If we visualise a function by having arrows from elements $a \in A$ to $f(a) \in B$ then we have the following:

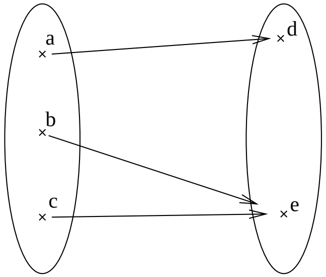
- A function is **injective**, if for every element of B there is **at most one arrow pointing to it**:



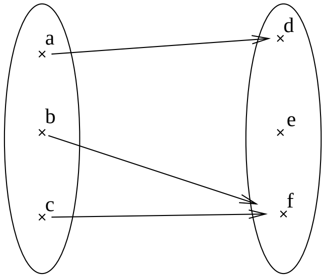
injective non-injective

Visualisation of “Surjective”

- A function is **surjective**, if for every element of B there is **at least one arrow pointing to it**:



surjective



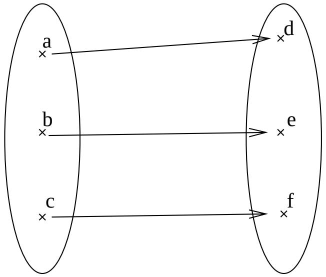
non-surjective

Remark

- The injective, surjective, bijective functions are closed under composition:
  - If $f : A \rightarrow B$ and $g : B \rightarrow C$ are injective (or surjective or bijective), then $g \circ f : A \rightarrow C$ is injective (surjective, bijective, respectively) as well.
- **Proof:** See mathematics lectures or easy exercise.

Visualisation of “Bijective”

- A function is **bijective**, if for every element of B there is **exactly one arrow pointing to it**:



bijective

- Note that, since we have a function, for every element of A there is exactly one arrow originating from there.

Infinite Sequences

- An infinite sequence of elements of a set B is an enumeration of certain elements of B by natural numbers.
  - E.g. the sequence of even numbers is

$$(0, 2, 4, 6, 8, \ldots)$$

  - We might repeat elements, e.g.

$$(0, 2, 0, 2, 0, 2, \ldots)$$

Infinite Sequences

- Sequences of natural numbers are written as

$$(a_n)_{n\in\mathbb{N}}$$

which stands for

$$(a_0, a_1, a_2, \ldots)$$

- So the sequence of even numbers is

$$\begin{aligned} &(0, 2, 4, 6, \ldots) \\ &= (a_0, a_1, a_2, \ldots) \\ &= (a_n)_{n\in\mathbb{N}} \end{aligned}$$

where

$$a_n = 2n$$

Infinite Sequences

- A sequence

$$(a_n)_{n\in\mathbb{N}}$$

of elements in A is nothing but a function $f : \mathbb{N} \to A$, s.t.

$$f(n) = a_n$$

.

- In fact we will identify functions $f : \mathbb{N} \to A$ with infinite sequences of elements of A.

Infinite Sequences

- So the following denotes the same mathematical object
  - The function $f : \mathbb{N} \to \mathbb{N}$, $f(n) = \begin{cases} 0 & \text{if } n \text{ is odd,} \\ 1 & \text{if } n \text{ is even.} \end{cases}$
  - The sequence $(1, 0, 1, 0, 1, 0, \ldots)$.
  - The sequence $(a_n)_{n\in\mathbb{N}}$ where $a_n = \begin{cases} 0 & \text{if } n \text{ is odd,} \\ 1 & \text{if } n \text{ is even.} \end{cases}$

Infinite Sequences

- Occasionally, we will enumerate sequences by different index sets.
  - E.g. we consider a sequence indexed by non-zero natural numbers

$$(a_n)_{n\in\mathbb{N}\setminus\{0\}}$$

or a sequence indexed by integers

$$(a_z)_{z\in\mathbb{Z}}$$

- A sequence $(a_x)_{x\in B}$ of elements in A is nothing but the function

$$f : B \to A \quad , \quad f(x) = a_x$$

λ-Notation

- $\lambda x.t$ means in an informal setting the function mapping x to t.
  E.g.
  - $\lambda x.x + 3$ is the function f s.t. $f(x) = x + 3$.
  - $\lambda x.\sqrt{x}$ is the function f s.t. $f(x) = \sqrt{x}$.
- This notation used, if one one wants to introduce a function without giving it a name.
- Domain and codomain not specified – when this notation is used, this will be clear from the context.

The "dot"-notation.

- In
  $$(A \rightarrow \forall x.B(x) \wedge C(x)) \vee D(x)$$
  $\forall x$ refers
  only to
  $$B(x) \wedge C(x)$$
  This is the maximum scope possible
  It doesn't make sense to include ") $\vee$ $D(x)$" into the scope.

The "dot"-notation.

- In expressions like
  $$\forall x.A(x) \wedge B(x)$$
  the quantifier ($\forall x.$) is as far as possible:
  - In
    $$\forall x.A(x) \wedge B(x)$$
    $\forall x.$ refers to
    $$A(x) \wedge B(x)$$

The "dot"-notation.

- In
  $$\exists x.A(x) \wedge B(x)$$
  $\exists x$ refers to
  $$A(x) \wedge B(x)$$
- In
  $$(A \wedge \exists x.B(x) \vee C(x)) \wedge D(x)$$
  $\exists x$ refers to
  $$B(x) \vee C(x)$$

The “dot”-notation.

- This applies as well to λ-expressions.
  - So

    $$\lambda x.x + x$$

    is the function taking an x and returning $x + x$.

Relations, Predicates and Sets

- A **predicate** on a set A is a property P of elements of A. In this lecture, A will usually be $\mathbb{N}^k$ for some $k \in \mathbb{N}$, $k > 0$.
- We write $P(a)$ for “predicate P is true for the element a of A”.
- We often write “$P(x)$ **holds**” for “$P(x)$ is true”.

Relations, Predicates and Sets

- We can use $P(a)$ in formulas. Therefore:
  - $\neg P(a)$ (“not $P(a)$”) means that “$P(a)$ is not true”.
  - $P(a) \wedge Q(b)$ means that “both $P(a)$ and $Q(b)$ are true”
  - $P(a) \vee Q(b)$ means that “$P(a)$ or $Q(b)$ is true”.

    (We have inclusive or: if both $P(a)$ and $Q(b)$ are true, then $P(a) \vee Q(b)$ is true as well).
  - $\forall x \in B.P(x)$ means that “for all elements x of the set B $P(x)$ is true”.
  - $\exists x \in B.P(x)$ means that “there exists an element x of the set B s.t. $P(x)$ is true”.

Relations, Predicates and Sets

- In this lecture, “relation” is another word for “predicate”.
- We identify a predicate P on a set A with $\{x \in A \mid P(x)\}$ Therefore predicates and sets will be identified. E.g., if P is a predicate,
  - $x \in P$ stands for $x \in \{x \in A \mid P(x)\}$, which is equivalent to $P(x)$,
  - $\forall x \in P.\varphi(x)$ for a formula φ stands for $\forall x.P(x) \rightarrow \varphi(x)$.
  - etc.

Relations, Predicates and Sets

- An n**-ary relation** or predicate on $\mathbb{N}$ is a relation $P \subseteq \mathbb{N}^n$.
  A **unary**, **binary**, **ternary** relation on $\mathbb{N}$ is a 1-ary, 2-ary, 3-ary relation on $\mathbb{N}$, respectively.
  - For instance $<$ and equality are binary relations on $\mathbb{N}$.
- An n**-ary function** on $\mathbb{N}$ is a function $f : \mathbb{N}^n \to \mathbb{N}$.
  A **unary**, **binary**, **ternary** function on $\mathbb{N}$ is a 1-ary, 2-ary, 3-ary function on $\mathbb{N}$, respectively.

$\vec{x}, \vec{y}$ etc.

- In many expressions we will have arguments, to which we don't refer explicitly.
  Example: Variables $x_0, \ldots, x_{n-1}$ in

$$f(x_0, \ldots, x_{n-1}, y) = \begin{cases} g(x_0, \ldots, x_{n-1}), & \text{if } y = 0, \\ h(x_0, \ldots, x_{n-1}), & \text{if } y > 0. \end{cases}$$

- We abbreviate $x_0, \ldots, x_{n-1}$, by $\vec{x}$.
- Then the above can be written shorter as

$$f(\vec{x}, y) = \begin{cases} g(\vec{x}), & \text{if } y = 0, \\ h(\vec{x}), & \text{if } y > 0. \end{cases}$$

- In general, $\vec{x}$ stands for $x_0, \ldots, x_{n-1}$, where the number of arguments n is clear from the context.

Examples

- If
  $$f : \mathbb{N}^{n+1} \to \mathbb{N}$$
  then in $f(\vec{x}, y)$,
  $\vec{x}$ needs to stand for n arguments.
  Therefore
  $$\vec{x} = x_0, \ldots, x_{n-1}$$
- If
  $$f : \mathbb{N}^{n+2} \to \mathbb{N}$$
  then in $f(\vec{x}, y)$,
  $\vec{x}$ needs to stand for $n + 1$ arguments,
  so
  $$\vec{x} = x_0, \ldots, x_n$$

Examples

- If P is an $n + 4$-ary relation, then in $P(\vec{x}, y, z)$,
  $\vec{x}$ stands for
  $$x_0, \ldots, x_{n+1}$$
- Similarly, we write $\vec{y}$ for
  $$y_0, \ldots, y_{n-1}$$
  where n is clear from the context.
- Similarly for
  $$\vec{z}, \vec{n}, \vec{m}, \ldots$$

Notation

- $$\forall \vec{x} \in \mathbb{N}.\varphi(\vec{x})$$

  stands for

  $$\forall x_0, \ldots, x_{n-1} \in \mathbb{N}.\varphi(x_0, \ldots, x_{n-1})$$

  where the number of variables n is implicit (and usually unimportant).

- $$\exists \vec{x} \in \mathbb{N}.\varphi(\vec{x})$$

  is to be understood similarly.

Notation

- $$\{\vec{x} \in \mathbb{N}^n \mid \varphi(\vec{x})\}$$

  is to be understood as

  $$\{(x_0, \ldots, x_{n-1}) \in \mathbb{N}^n \mid \varphi(x_0, \ldots, x_{n-1})\}$$

- $$\{(\vec{x}, y, z) \in \mathbb{N}^{n+2} \mid \varphi(\vec{x}, y, z)\}$$

  is to be understood as

  $$\{(x_0, \ldots, x_{n-1}, y, z) \in \mathbb{N}^{n+2} \mid \varphi(x_0, \ldots, x_{n-1}, y, z)\}$$

- Similar notations are to be understood analogously.

(b) Cardinality

- In this subsection, we will make precise the notion of “small”, “big” sets above.
- So we need a notion of size of a set.
- For finite sets one can introduce a number for the size of a set.
- For infinite sets, introducing such numbers (cardinality) is beyond the scope of this lectures
- However, we can introduce a notion of **relative size**, namely what it means for one set to be **smaller/equal/greater in size** than another set.
  - **Equinumerous** will mean “equal in size”.

Number of Elements

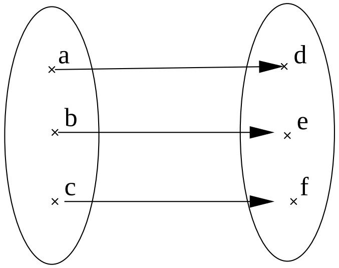
Notation 2.3
If A is a finite set, let $|A|$ be the number of elements in A.

Remark 2.4

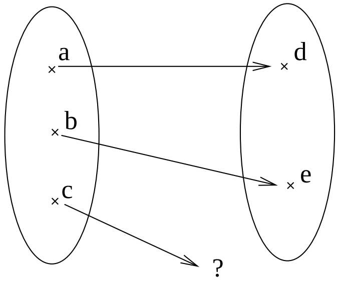
One sometimes writes $\#A$ for $|A|$.

Cardinality of Finite Sets

If A and B are finite sets, then $|A| = |B|$, if and only if there is a bijection between A and B:

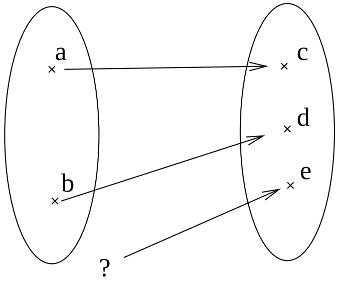


Bijection exists



No Bijection

Cardinality of Finite Sets



No Bijection

- The above can be generalized to arbitrary (possibly infinite sets) as follows:

Cardinality of Sets

Definition 2.5
Two sets A and B are **equinumerous** *or* "**have the same cardinality**"*, in mathematical notation*

$$A \approx B$$

if there exists a bijection

$$f : A \to B$$

Remark 2.6
If A and B are finite sets, then $A \approx B$ if and only A and B have the same number of elements, i.e. $|A| = |B|$.

Cardinality of infinite sets

- However we have $\mathbb{N}$ and $\mathbb{N} \cup \{\bullet\}$, where $\bullet$ is a new element, are equinumerous.
  - $f : \mathbb{N} \to \mathbb{N} \cup \{\bullet\}$, s.t.
    - $f(0) = \bullet$, $f(n+1) = n$

    is a bijection.
- Analogy with a hotel with infinite many rooms numbered by natural numbers.
  - This hotel can always accomodate a new guest, by moving every guest from room n to room $n+1$, and the new guest to room no. 0.

Change of Notation

- Until the academic year 2004/05, we used in lectures
  - “have the same cardinality” instead of “equinumerous”,
  - and $\simeq$ instead of $\approx$.
    - Note that $\simeq$ is used (and was used) for partial equality as well.
    - Change of notation in order to avoid the overloading of notation.
  - Please take this into account when looking at old exams and other lecture material.
- Both notions occur as well in the literature and might be used in other modules.

$\approx$ as an Equivalence Relation

Lemma 2.7
$\approx$ is an equivalence relation, i.e. for all sets A, B, C we have:

(a) **Reflexivity.** $A \approx A$.

(b) **Symmetry.** *If $A \approx B$, then $B \approx A$.*

(c) **Transitivity.** *If $A \approx B$ and $B \approx C$, then $A \approx C$.*

Proof:
(a): The function $\mathrm{id} : A \to A$, $\mathrm{id}(a) = a$ is a bijection.
(b): If $f : A \to B$ is a bijection, so is its inverse f^{-1}.

(c): If $f : A \to B$ and $g : B \to C$ are bijections, so is the

composition $g \circ f : A \to C$.

Notion of Cardinality in Set Theory

- In set theory there exists the notion of a **cardinality**, which is some kind of number (an **ordinal**) which measures the size of a set.
  - Then one can show:
    - $A \approx B$ iff the cardinality expressed as an ordinal for A and B is the same.
  - However, this notion is beyond the scope of this module.

Meaning of the above

- That $\approx$ is an equivalence relation means that it has properties we expect of a relation expressing that two sets have the same size:
  - Every set has the same size as itself
$$A \approx A$$
  - If A has the same size as B, then B has the same size as A.
$$A \approx B \to B \approx A$$
  - If A has the same size as B and B has the same size as C then A has the same size as C:
$$(A \approx B \land B \approx C) \to A \approx C$$

Meaning of the above

- If we wrongly defined A and B to have the same size if there is an injection from A to B then symmetry wouldn't hold.
- So there is something to be shown, the language notation we use only suggests that the above mentioned properties hold.
  - Don't let yourself be deceived by language!

Stronger Result

- In fact we will show something even stronger:
  For any set A the following holds:
  there is no surjection
  $$C : A \to \mathcal{P}(A)$$
- If this is shown, then we know that there is no bijection $C : A \to \mathcal{P}(A)$, $A \not\approx \mathcal{P}(A)$.
- **Remark on Notation:**
  - We write here the capital letter C instead of the usual letters f, g etc. for functions, in order to flag that $C(a)$ is a set.
  - For notational convenience we write C_a instead of $C(a)$, so C_a is "the ath set enumerated by the function C".

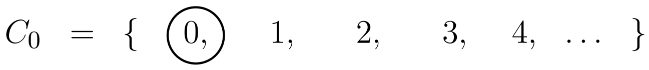
Cardinality of the Power Set

Theorem 2.8
A set A and its power set $\mathcal{P}(A) := \{B \mid B \subseteq A\}$ are never equinumerous:
$$A \not\approx \mathcal{P}(A)$$

Proof

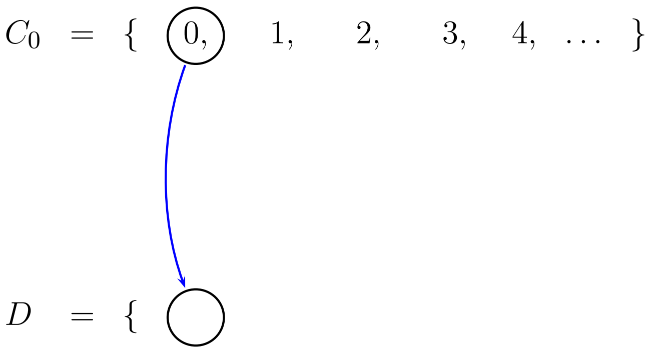
- A typical diagonalisation argument.
- First consider the case $A = \mathbb{N}$.
- Assume $C : \mathbb{N} \to \mathcal{P}(\mathbb{N})$ is a surjection.
- We define a set $D \subseteq \mathbb{N}$ s.t. $D \neq C_n$ for every $n \in \mathbb{N}$.
- $D = C_n$ will be violated at element n:
  - If $n \in C_n$, we add n not to D, therefore $n \in C_n \wedge n \notin D$.
  - If $n \notin C_n$, we add n to D, therefore $n \notin C_n \wedge n \in D$.
- On the next slide we take as an example some function $C : \mathbb{N} \to \mathcal{P}(\mathbb{N})$ and show how to construct a set D s.t. $C_n \neq D$ for all $n \in \mathbb{N}$.

Example



$C_0 = \{ 0, \quad 1, \quad 2, \quad 3, \quad 4, \quad \ldots \}$

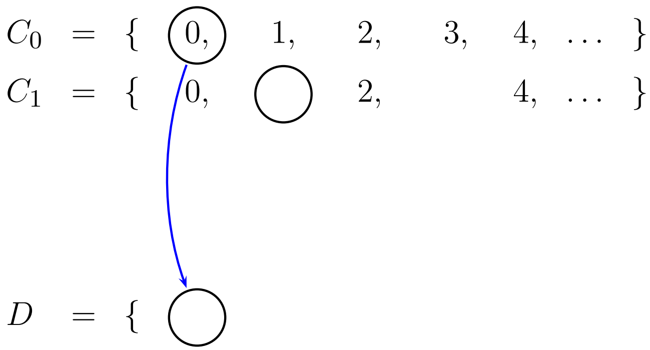
Example



$C_0 = \{ 0, \quad 1, \quad 2, \quad 3, \quad 4, \quad \ldots \}$

$D = \{$

Example

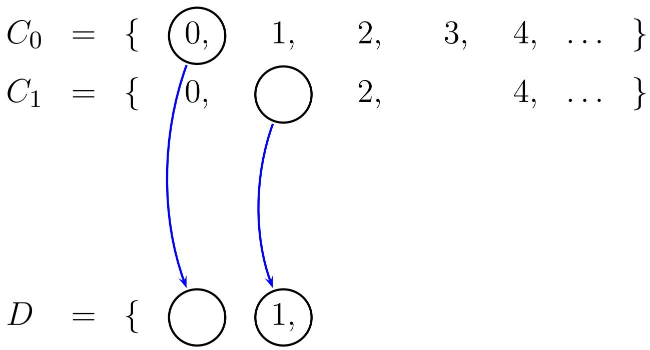


$C_0 = \{ 0, \quad 1, \quad 2, \quad 3, \quad 4, \quad \ldots \}$

$C_1 = \{ 0, \quad 2, \quad 4, \quad \ldots \}$

$D = \{$

Example

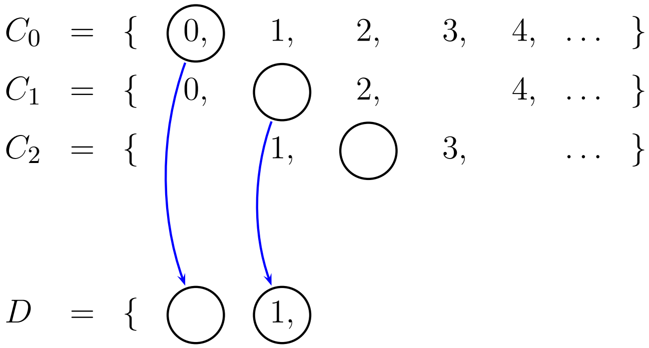


$C_0 = \{ 0, \quad 1, \quad 2, \quad 3, \quad 4, \quad \ldots \}$

$C_1 = \{ 0, \quad 2, \quad 4, \quad \ldots \}$

$D = \{ \quad 1,$

Example



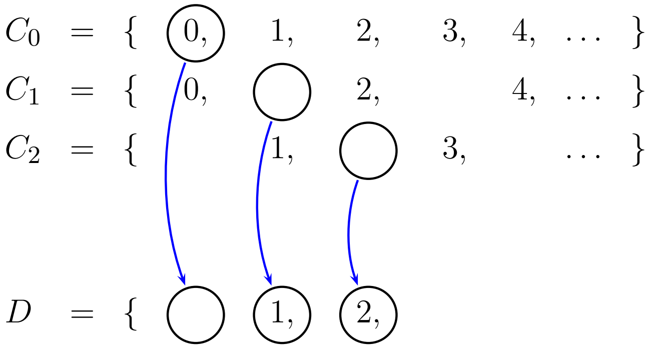
$C_0 = \{ \ 0, \ 1, \ 2, \ 3, \ 4, \ \ldots \ \}$

$C_1 = \{ \ 0, \ \bigcirc \ 2, \ 4, \ \ldots \ \}$

$C_2 = \{ \ 1, \ \bigcirc \ 3, \ \ldots \ \}$

$D = \{ \ \bigcirc \ 1,$

Example



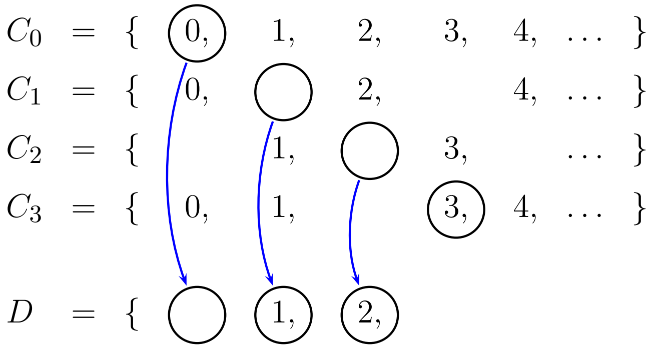
$C_0 = \{ \ 0, \ 1, \ 2, \ 3, \ 4, \ \ldots \ \}$

$C_1 = \{ \ 0, \ \bigcirc \ 2, \ 4, \ \ldots \ \}$

$C_2 = \{ \ 1, \ \bigcirc \ 3, \ \ldots \ \}$

$D = \{ \ \bigcirc \ 1, \ 2,$

Example



$C_0 = \{ \ 0, \ 1, \ 2, \ 3, \ 4, \ \ldots \ \}$

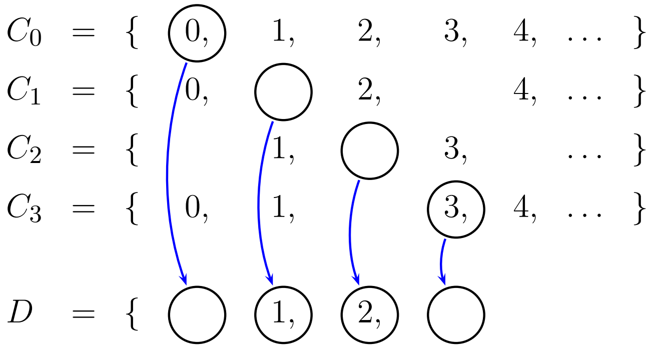
$C_1 = \{ \ 0, \ \bigcirc \ 2, \ 4, \ \ldots \ \}$

$C_2 = \{ \ 1, \ \bigcirc \ 3, \ \ldots \ \}$

$C_3 = \{ \ 0, \ 1, \ 3, \ 4, \ \ldots \ \}$

$D = \{ \ \bigcirc \ 1, \ 2,$

Example



$C_0 = \{ \ 0, \ 1, \ 2, \ 3, \ 4, \ \ldots \ \}$

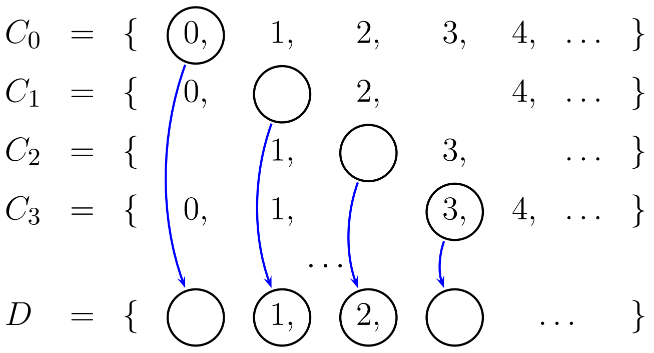
$C_1 = \{ \ 0, \ \bigcirc \ 2, \ 4, \ \ldots \ \}$

$C_2 = \{ \ 1, \ \bigcirc \ 3, \ \ldots \ \}$

$C_3 = \{ \ 0, \ 1, \ 3, \ 4, \ \ldots \ \}$

$D = \{ \ \bigcirc \ 1, \ 2, \ \bigcirc$

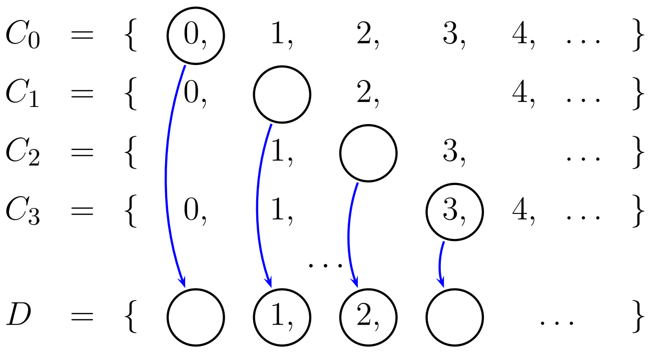
Example



$$\begin{array}{lcllllllll}
C_0 & = & \{ & \bigcirc\!\!\!\!\!\!0, & 1, & 2, & 3, & 4, & \ldots & \} \\
C_1 & = & \{ & 0, & \bigcirc & 2, & & 4, & \ldots & \} \\
C_2 & = & \{ & & 1, & \bigcirc & 3, & & \ldots & \} \\
C_3 & = & \{ & 0, & 1, & & \bigcirc\!\!\!\!\!\!3, & 4, & \ldots & \} \\
 & & & & & \ldots & & & & \\
D & = & \{ & \bigcirc & \bigcirc\!\!\!\!\!\!1, & \bigcirc\!\!\!\!\!\!2, & \bigcirc & & \ldots & \}
\end{array}$$

We were going through the diagonal in the above matrix.

Example



$$\begin{array}{lcllllllll}
C_0 & = & \{ & \bigcirc\!\!\!\!\!\!0, & 1, & 2, & 3, & 4, & \ldots & \} \\
C_1 & = & \{ & 0, & \bigcirc & 2, & & 4, & \ldots & \} \\
C_2 & = & \{ & & 1, & \bigcirc & 3, & & \ldots & \} \\
C_3 & = & \{ & 0, & 1, & & \bigcirc\!\!\!\!\!\!3, & 4, & \ldots & \} \\
 & & & & & \ldots & & & & \\
D & = & \{ & \bigcirc & \bigcirc\!\!\!\!\!\!1, & \bigcirc\!\!\!\!\!\!2, & \bigcirc & & \ldots & \}
\end{array}$$

We were going through the diagonal in the above matrix.

Therefore this proof is called a **diagonalisation argument.**

Proof

So we define

$$D := \{n \in \mathbb{N} \mid n \notin C_n\} \ .$$

We have $D \neq C_n$ for all n:
Assume $D = C_n$.

- If $n \in D$, then by the definition of D we have $n \notin C_n$, therefore by $D = C_n$ we get $n \notin D$, a contradiction.
- If $n \notin D$, then by the definition of D we have $n \in C_n$, therefore by $D = C_n$ we get $n \in D$, a contradiction.

Therefore we obtain a contradiction in both cases, $D \neq C_n$.

Therefore D is not in the image of C, so C is not a surjection

a **contradiction.**

Formal Proof ($A = \mathbb{N}$)

In short, the above argument for $A = \mathbb{N}$ reads as follows:
Assume $C : \mathbb{N} \to \mathcal{P}(\mathbb{N})$ is a surjection.
Define

$$D := \{n \in \mathbb{N} \mid n \notin C_n\} \ .$$

Since C is surjective, D must be in the image of C.
Assume $D = C_n$.
Then we have

$$\begin{array}{rcl}
n \in D & \overset{\text{Definition of } D}{\Leftrightarrow} & n \notin C_n \\
 & \overset{D=C_n}{\Leftrightarrow} & n \notin D \\
 & & \text{a contradiction}
\end{array}$$

General Situation

For general A, the proof is almost identical:
Assume $C : A \to \mathcal{P}(A)$ is a surjection.
We define a set D, s.t. $D = C_a$ is violated for a:

$$D := \{a \in A \mid a \notin C_a\}$$

Since C is surjective, D must be in the image of C.
Assume $D = C_a$. Then we have

$$\begin{array}{lcl} a \in D & \overset{\text{Definition of } D}{\Leftrightarrow} & a \notin C_a \\ & \overset{D=C_a}{\Leftrightarrow} & a \notin D \\ & & \text{a contradiction} \end{array}$$

Proof

- Let for $B \in \mathcal{P}(A)$

$$\begin{array}{rcl} \chi_B & : & A \to \{0,1\} \\ \chi_B(x) & := & \begin{cases} 1 & \text{if } x \in B, \\ 0 & \text{if } x \notin B. \end{cases} \end{array}$$

χ_B is called the **characteristic function of B**.

- If we consider 0 as false and 1 as true, then we get

$$\chi_B(x) = \begin{cases} \mathrm{true} & \text{if } x \in B, \\ \mathrm{false} & \text{if } x \notin B. \end{cases}$$

- Therefore χ_B is the function, which determines whether its argument is in B or not.

$\mathcal{P}(A)$ and $A \to \mathrm{Bool}$

Lemma 2.9 For every set A

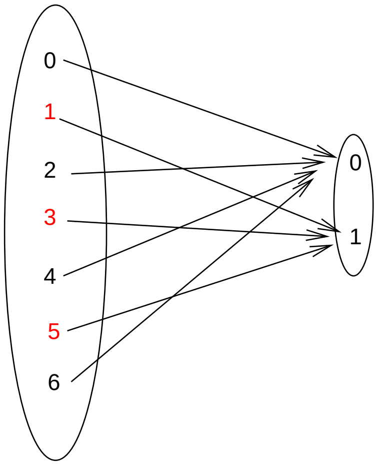
$$\mathcal{P}(A) \approx (A \to \mathrm{Bool}) \approx (A \to \{0,1\})$$

Remark: Note that we can identify the set of Booleans Bool with $\{0,1\}$ by identifying

- true with 1,
- false with 0.

Therefore we get $(A \to \mathrm{Bool}) \approx (A \to \{0,1\})$.

Example: B = set of Odd Number



$$\chi_B(n) = \begin{cases} 0 & \text{if } n \text{ is even,} \\ 1 & \text{if } n \text{ is odd.} \end{cases}$$

Proof

- χ is a function from $\mathcal{P}(A)$ to $A \to \{0,1\}$, where we write the application of χ to an element B as χ_B instead of $\chi(B)$.
- We show that χ is a bijection.
  - Then it follows that $\mathcal{P}(A) \approx (A \to \{0,1\})$.

    Jump over rest of proof

$$\chi_B(x) := \begin{cases} 1 & \text{if } x \in B, \\ 0 & \text{if } x \notin B. \end{cases}$$

- χ has an inverse:
  Define

$$\begin{array}{rcl} \chi^{-1} & : & (A \to \{0,1\}) \to \mathcal{P}(A) \\ \chi^{-1}(f) & := & \{x \in A \mid f(x) = 1\} \end{array}$$

χ and χ^{-1} are inverse

$$\begin{array}{rcl} \chi_B(x) & := & \begin{cases} 1 & x \in B, \\ 0 & \text{otherwise.} \end{cases} \\ \chi^{-1}(f) & := & \{x \in A \mid f(x) = 1\} \end{array}$$

- We show that χ and χ^{-1} are inverse:
- $\chi^{-1} \circ \chi$ is the identity:
- If $B \subseteq A$, then

$$\begin{array}{rcl} \chi^{-1}(\chi_B) & = & \{x \in A \mid \chi_B(x) = 1\} \\ & = & \{x \in A \mid x \in B\} \\ & = & B \end{array}$$

χ and χ^{-1} are inverse

$$\begin{array}{rcl} \chi_B(x) & := & \begin{cases} 1 & x \in B, \\ 0 & \text{otherwise.} \end{cases} \\ \chi^{-1}(f) & := & \{x \in A \mid f(x) = 1\} \end{array}$$

- $\chi \circ \chi^{-1}$ is the identity:
- If $f : A \to \{0,1\}$, then

$$\begin{array}{rcl} \chi_{\chi^{-1}(f)}(x) = 1 & \Leftrightarrow & x \in \chi^{-1}(f) \\ & \Leftrightarrow & f(x) = 1 \end{array}$$

χ and χ^{-1} are inverse

$$\begin{aligned} \chi_B(x) &:= \begin{cases} 1 & x \in B, \\ 0 & \text{otherwise.} \end{cases} \\ \chi^{-1}(f) &:= \{x \in A \mid f(x) = 1\} \end{aligned}$$

and

$$\begin{aligned} \chi_{\chi^{-1}(f)}(x) = 0 \quad &\Leftrightarrow \quad x \not\in \chi^{-1}(f) \\ &\Leftrightarrow \quad f(x) \neq 1 \\ &\Leftrightarrow \quad f(x) = 0 \quad . \end{aligned}$$

Therefore $\chi_{\chi^{-1}(f)} = f$.

χ and χ^{-1} are inverse

It follows that χ is bijective and therefore

$$\mathcal{P}(A) \approx (A \to \{0, 1\}) \quad .$$

(c) Countable Sets

Definition 2.10

- *A set A is* **countable***, if it is finite or $A \approx \mathbb{N}$.*
- *A set, which is not countable, is called* **uncountable***.*

- Intuitively
  - uncountable sets are very big
  - countable sets are finite or small infinite sets.
    - Countable sets have at most the size of the $\mathbb{N}$.

Relationship to Cardinality

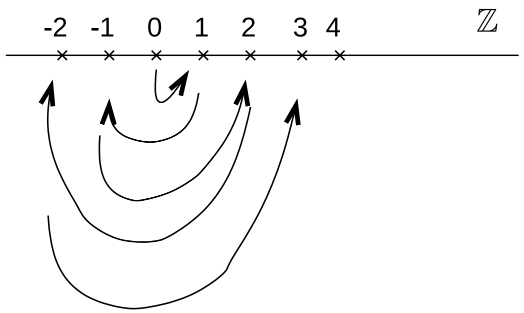
- Intuitively (this can be made mathematically precise) the cardinalities of sets start with the finite cardinalities $0, 1, 2, \ldots$ corresponding to finite sets having $0, 1, 2, \ldots$ elements.
  - All these cardinalities are different (for finite sets A, B we have $A \approx B$ iff A and B have the same number of elements).
- Then the next cardinality is that of $\mathbb{N}$.
- Then we have higher cardinalities like the cardinality of $\mathcal{P}(\mathbb{N})$ (or $\mathbb{R}$).

Relationship to Cardinality

$$\begin{array}{c} 0 \\ 1 \\ 2 \\ \ldots \\ \mathbb{N} \\ \mathcal{P}(\mathbb{N}) \\ \ldots \end{array}$$

- Countable sets are the sets having cardinality less than or equal the cardinality of $\mathbb{N}$.
  - Which means they have cardinality of $\mathbb{N}$ or finite cardinality.

Illustration of $\mathbb{Z}$ is Countable



Examples of (Un)countable Sets

- $\mathbb{N}$ is countable.
- $\mathbb{Z} := \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ is countable.
  - We can enumerate the elements of $\mathbb{Z}$ in the following way:
    $0, +1, -1, +2, -2, +3, -3, +4, -4, \ldots$.
    So we have the following map:
    $0 \mapsto 0,\ \ 1 \mapsto +1,\ \ 2 \mapsto -1,\ \ 3 \mapsto +2,\ \ 4 \mapsto -2$, etc.
    This map can be described as follows:
    $g : \mathbb{N} \to \mathbb{Z}$,

$$g(n) := \begin{cases} \frac{-n}{2} & \text{if } n \text{ is even,} \\ \frac{n+1}{2} & \text{if } n \text{ is odd.} \end{cases}$$

Exercise: Show that g is bijective.

Examples of (Un)countable Sets

- $\mathcal{P}(\mathbb{N})$ is uncountable.
  - $\mathcal{P}(\mathbb{N})$ is not finite.
  - $\mathbb{N} \not\approx \mathcal{P}(\mathbb{N})$.
- $\mathcal{P}(\{1, \ldots, 10\})$ is countable.
  - Since it is finite.

Characterisation of Countable Sets

Lemma 2.11

A set A is countable, if and only if there is an injective map $g : A \to \mathbb{N}$.

Remark 2.12

Intuitively, Lemma 2.11 expresses: A is countable, if we can assign to every element $a \in A$ a unique code $f(a) \in \mathbb{N}$. However, it is not required that each element of $\mathbb{N}$ occurs as a code.

The code $f(a)$ can be considered as a finite description of a. So A is countable if we can give a unique finite description for each of its element.

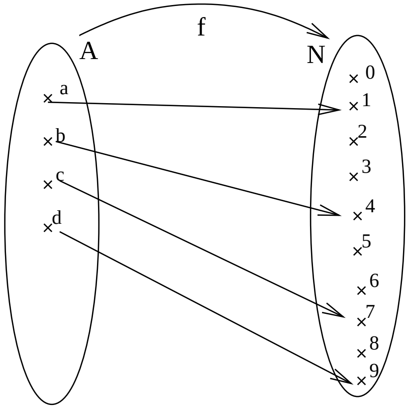
Proof of Lemma 2.11, “⇒”

“⇒”:
Assume A is countable.
Show that there exists an injective function $f : A \to \mathbb{N}$.

- Case A is finite:
  Let $A = \{a_0, \ldots, a_n\}$, where a_i are different.
  We can define $f : A \to \mathbb{N}$, $a_i \mapsto i$.
  f is injective.
- Case A is infinite:
  A is countable, so there is a bijection from A into $\mathbb{N}$, which is therefore injective.

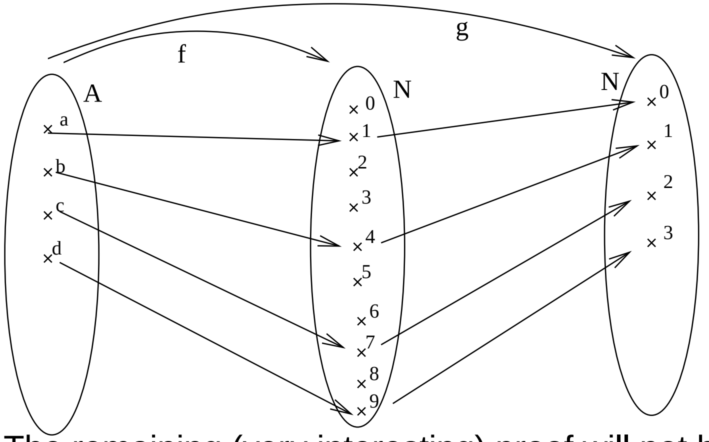
Proof of Lemma 2.11, “⇐”

“⇐”: Assume $f : A \to \mathbb{N}$ is injective.
Show A is countable.
If A is finite, we are done.
Assume A is infinite. Then f is for instance something like the following:



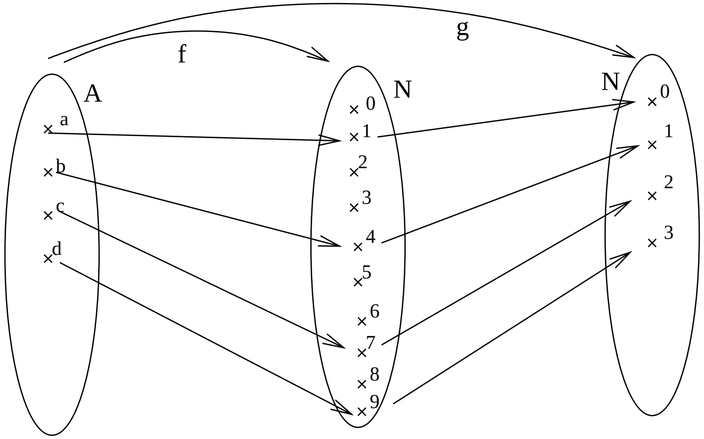
Proof of Lemma 2.11, “⇐”

In order to obtain a bijection $g : A \to \mathbb{N}$, we need to jump over the gaps in the image of f:



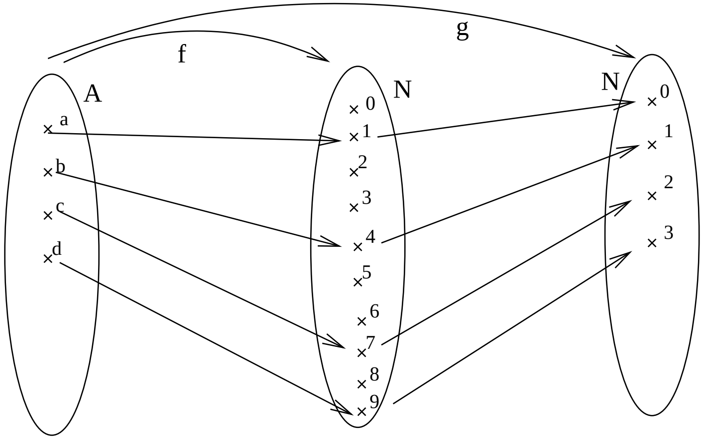
The remaining (very interesting) proof will not be given in the lecture. Jump over remaining proof.

Proof of Lemma 2.11, “$\Leftarrow$”



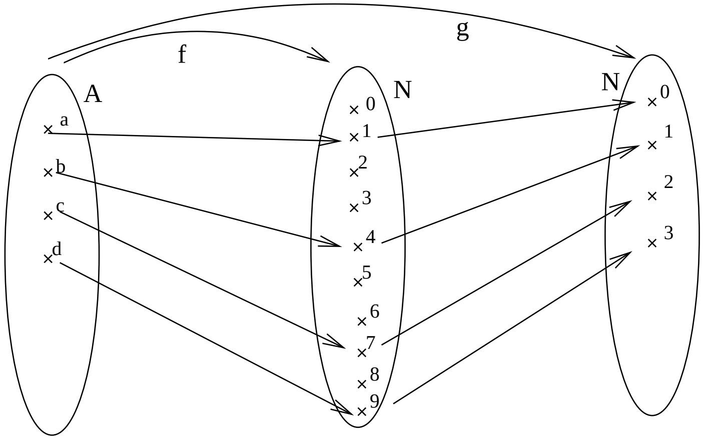
- $f(a) = 1$, which is the element number 0 in the image of f.
  g should instead map a to 0.

Proof of Lemma 2.11, “$\Leftarrow$”



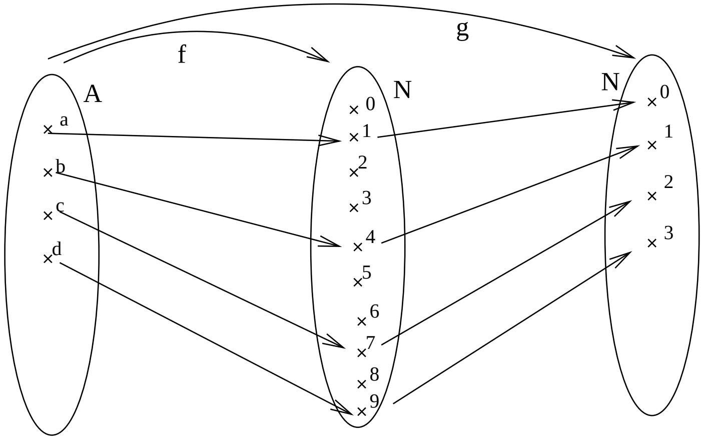
- 1 is element number 0 in the image of f, because the number of elements $f(a')$ below $f(a)$ is 0.

Proof of Lemma 2.11, “$\Leftarrow$”



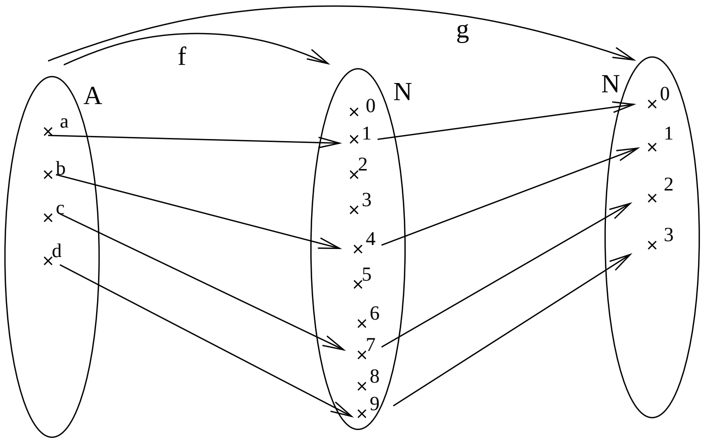
- $f(b) = 4$, which is the element number 1 in the image of f.
  g should instead map b to 1. Etc.

Proof of Lemma 2.11, “$\Leftarrow$”



- 4 is element number 1 in the image of f, because the number of elements $f(a')$ below $f(b)$ is 1.

Proof of Lemma 2.11, “$\Leftarrow$”



So in general we define $g : A \to \mathbb{N}$.

$$g(a) := |\{a' \in A \mid f(a') < f(a)\}|$$

Proof of Lemma 2.11, “$\Leftarrow$”

$$g(a) = |\{a' \in A \mid f(a') < f(a)\}|$$

We show that g is a bijection:

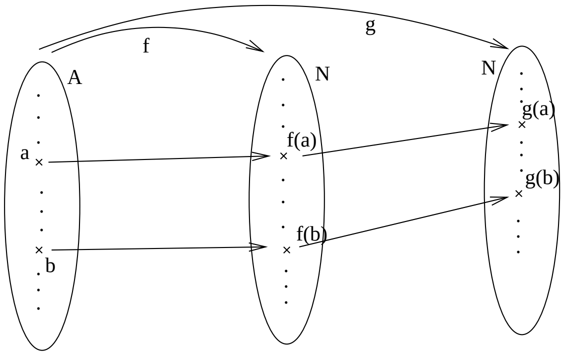
- g is injective:
  Assume $a, b \in A$, $a \neq b$.
  Show $g(a) \neq g(b)$.
  By the injectivity of f we have $f(a) \neq f(b)$.
  Let for instance $f(a) < f(b)$.

Proof of Lemma 2.11, “$\Leftarrow$”

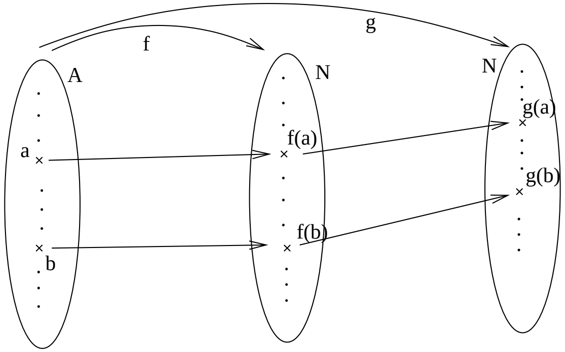
$$g(a) := |\{a' \in A \mid f(a') < f(a)\}|$$

g is well defined, since f is injective, so the number of $a' \in A$ s.t. $f(a') < f(a)$ is finite.

Proof of Lemma 2.11, “$\Leftarrow$”



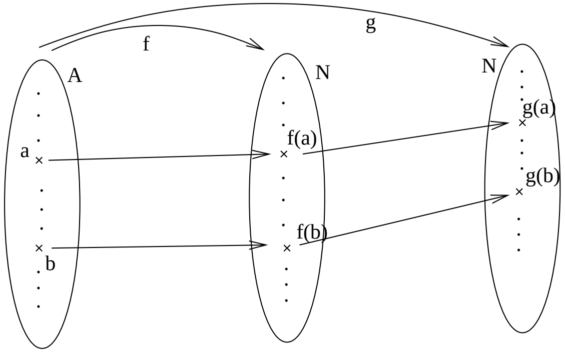
Proof of Lemma 2.11, “$\Leftarrow$”



Then

$$\{a' \in A \mid f(a') < f(a)\} \overset{\subset}{\neq} \{a' \in A \mid f(a') < f(b)\} \ ,$$

Proof of Lemma 2.11, “$\Leftarrow$”

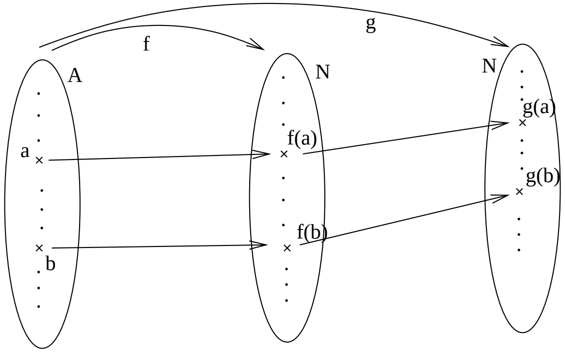


therefore

$$g(a) = |\{a' \in A \mid f(a') < f(a)\}| < |\{a' \in A \mid f(a') < f(b)\}| = g$$

$g(a) \neq g(b)$.

Proof of Lemma 2.11, “$\Leftarrow$”



therefore

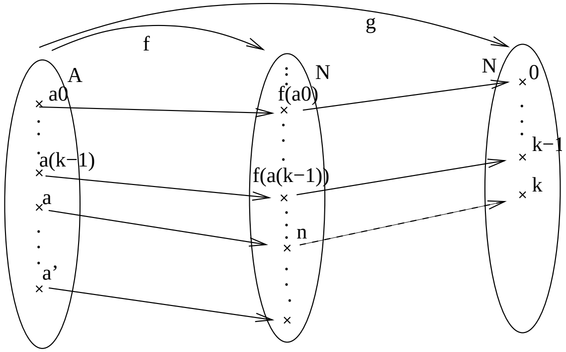
$$g(a) = |\{a' \in A \mid f(a') < f(a)\}| < |\{a' \in A \mid f(a') < f(b)\}| = g(b) \ ,$$

Proof of Lemma 2.11, “$\Leftarrow$”

$$g(a) = |\{a' \in A \mid f(a') < f(a)\}|$$

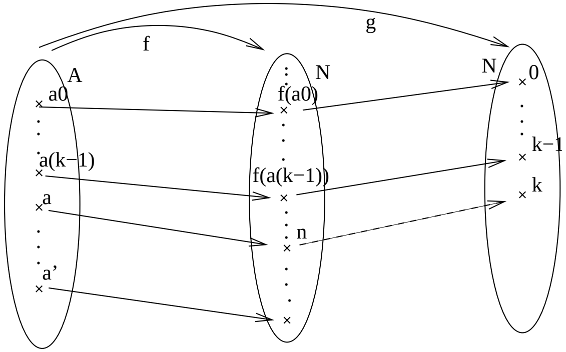
- g is surjective:
  We define by induction on k for $k \in \mathbb{N}$ an element $a_k \in A$
  s.t. $g(a_k) = k$. Then the assertion follows:
  Assume we have defined already $a_0, \ldots, a_{k-1}$.

Proof of Lemma 2.11, “$\Leftarrow$”



There exist infinitely many $a' \in A$, f is injective, so there must be at least one $a' \in A$ s.t. $f(a') > f(a_{k-1})$.

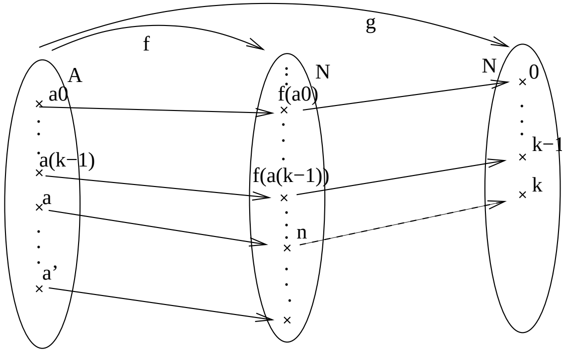
Proof of Lemma 2.11, “$\Leftarrow$”



n minimal s.t. $n = f(a')$ for some $a' \in A$, $n > f(a_{k-1})$

Let a be the unique element of A s.t. $f(a) = n$.

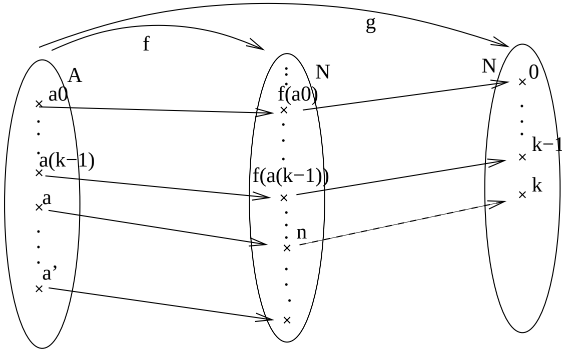
Proof of Lemma 2.11, “$\Leftarrow$”



There exists $a' \in A$ s.t. $f(a') > f(a_{k-1})$.

Let n be minimal s.t. $n = f(a)$ for some $a \in A$ and $n > f(a_{k-1})$.

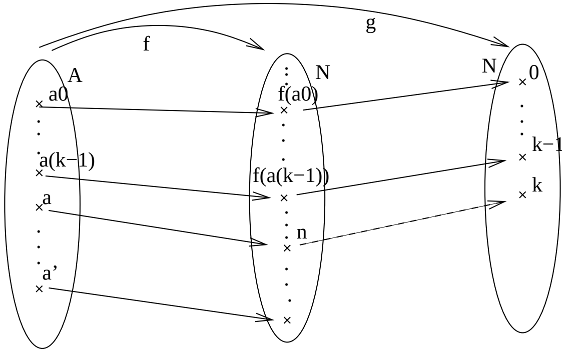
Proof of Lemma 2.11, “$\Leftarrow$”



n minimal s.t. $n = f(a)$ for some $a \in A$, $n > f(a_{k-1})$
$f(a) = n$

$$\{a'' \in A \mid f(a'') < f(a)\} = \{a'' \in A \mid f(a'') < f(a_{k-1})\} \cup \{a_{k-1}\}$$

Proof of Lemma 2.11, “$\Leftarrow$”



$$\begin{aligned}\text{Therefore } g(a) &= |\{a'' \in A \mid f(a'') < f(a)\}| \\ &= |\{a'' \in A \mid f(a'') < f(a_{k-1})\}| + 1 \\ &= g(a_{k-1}) + 1 = k - 1 + 1 = k \ .\end{aligned}$$

Let $a_k := a$.

Corollary

Corollary 2.13

(a) *If B is countable and $g : A \to B$ injective, then A is countable.*

(b) *If A is uncountable and $g : A \to B$ injective, then B is uncountable.*

(c) *If B is countable and $A \subseteq B$, then A is countable.*

Proof:

- (a) If B is countable, there exists an injection $f : B \to \mathbb{N}$. But then $f \circ g : A \to \mathbb{N}$ is an injection as well, therefore A is countable.
- (b): By (a). Why? (Exercise).
- (c): By (a). (What is g?; exercise).

Corollary (Cont.)

Corollary 2.13

(d) *If A is uncountable and $A \subseteq B$, then B is uncountable.*

(e) *If $A \approx B$, then A is countable if and only if B is countable.*

Proof:

- (d): By (c). Why? (Exercise).
- (e): By (a). Why ?

Remark:

- A corollary is a lemma/theorem which is a direct consequence of a more difficult lemma or theorem shown before.

Injection and Size

- Intuitively we can say:
  - That there exists an injective function

    $$f : A \to B$$

    means that the size of A is less than or equal to the size of B.
  - That $A \subseteq B$ means that there is an injection from A into B.
    - So the size of A is less than or equal to the size of B.

Characterisation of Count. Sets, II

Lemma 2.14
A set A is countable, if and only if $A = \emptyset$ or there exists a surjection $h : \mathbb{N} \to A$.

Remark: This explains the notion "countable": A non-empty set is countable if we can enumerate its elements (repetitions are allowed).

2nd Remark: The empty set $\emptyset$ is countable, but there exists no surjection $h : \mathbb{N} \to \emptyset$ – in fact there exists no function $h : \mathbb{N} \to \emptyset$ at all.

Jump over Proof.

Proof of Lemma 2.14

"$\Rightarrow$": Assume A is countable. If A is empty we are done.
So assume A is non-empty.
Show there exists a surjection $f : \mathbb{N} \to A$.

- Case A is finite.
  Assume $A = \{a_0, \ldots, a_n\}$.
  Define $f : \mathbb{N} \to A$,

$$f(k) := \begin{cases} a_k & \text{if } k \leq n \quad , \\ a_0 & \text{otherwise} \; . \end{cases}$$

  f is clearly surjective.

Proof of Lemma 2.14

- Case A is infinite.
  A is countable, so there exists a bijection from $\mathbb{N}$ to A, which is therefore surjective.

Proof of Lemma 2.14

"$\Leftarrow$":

- If $A = \emptyset$, then A is countable.
- So assume A and

$$h : \mathbb{N} \to A \text{ is surjective}$$

- Show A is countable.
- Define

$$\begin{array}{rcl} g & : & A \to \mathbb{N} \; , \\ g(a) & := & \min\{n \mid h(n) = a\} \; . \end{array}$$

- $g(a)$ is well-defined, since h is surjective:
  - There exists some n s.t. $h(n) = a$, therefore the minimal such n is well-defined.

Proof of Lemma 2.14

$$\begin{aligned} g \quad &: \quad A \to \mathbb{N} \ , \\ g(a) \quad &:= \quad \min\{n \mid h(n) = a\} \end{aligned}$$

- It follows that for $a \in A$ we have

$$h(g(a)) = a \ .$$

- Therefore g is injective:
  - If $g(a) = g(a')$ then

$$a = h(g(a)) = h(g(a')) = a' \ .$$

- Therefore $g : A \to \mathbb{N}$ is an injection, and by Lemma 2.11, A is countable.

Corollary

Corollary 2.15

(a) *If A is countable and $g : A \to B$ surjective, then B is countable.*

(b) *If B is uncountable and $g : A \to B$ surjective, then A is uncountable.*

Proof of Corollary 2.15 (a)

- To be shown: If A is countable, $g : A \to B$ is surjective, then B is countable as well.
- So assume A is countable, $g : A \to B$ is surjective.
- If A is empty, then B is empty as well and therefore countable.
  - (We need to treat $A = \emptyset$ as a special case, since in that case there exists no surjection $f : \mathbb{N} \to A$ as assumed in the next step, even so A is countable).

Proof of Corollary 2.15 (a)

- Otherwise there exists a surjection

$$f : \mathbb{N} \to A$$

But then

$$g \circ f : \mathbb{N} \to B$$

is a surjection as well,
therefore B is countable.

Proof of Corollary 2.15 (b)

- Follows by (a). Why?

Surjectivion and Size

- Intuitively we can say:
  - That there exists a surjective function
$$f : A \to B$$
    means that the size of A is greater than or equal to the size of B.

Examples of Uncountable Sets

Lemma 2.16
The following sets are uncountable:

(a) $F := \{f \mid f : \mathbb{N} \to \{0, 1\}\}$.

(b) $G := \{f \mid f : \mathbb{N} \to \mathbb{N}\}$.

(c) The set of real numbers $\mathbb{R}$.

- **Proof of (a):** By Lemma 2.9 $\mathcal{P}(\mathbb{N}) \approx (\mathbb{N} \to \{0, 1\})$. $\mathcal{P}(\mathbb{N})$ is uncountable, therefore $\mathbb{N} \to \{0, 1\}$ as well.
- **Proof of (b):** $F \subseteq G$, F is uncountable, so G is uncountable.

Idea of Proof of Lemma 2.16 (c)

- In order to show $\mathbb{R}$ is uncountable, it suffices to show that the half open interval $[0, 1[$
  (i.e. $\{x \in \mathbb{R} \mid 0 \leq x < 1\}$)
  is uncountable).
- Elements of $[0, 1[$ are in binary representation of the form
$$(0.a_0 a_1 a_2 a_3 \cdots)_2$$
  where $a_i \in \{0, 1\}$.
- $(a_n)_{n \in \mathbb{N}}$ is a function $\mathbb{N} \to \{0, 1\}$.
- If the function mapping sequences $(a_n)_{n \in \mathbb{N}} : \mathbb{N} \to \{0, 1\}$ to $\mathbb{R}$ were injective, then we could conclude from $\mathbb{N} \to \{0, 1\}$ uncountable that $[0, 1[$ and therefore $\mathbb{R}$ are uncountable.

Idea of Proof of Lemma 2.16 (c)

- However this function is not injective since $0.a_0a_1a_2\cdots a_n011111\cdots$ and $0.a_0a_1a_2\cdots a_n100000\cdots$ are the same number.
  - This is similar to decimal representation, where $0.a_0a_1a_2\cdots a_n099999\cdots$ and $0.a_0a_1a_2\cdots a_n100000\cdots$ are the same.
- This problem can be overcome with some effort.
- The detailed proof will be omitted in the lecture.
  Jump over Proof.

Proof of Lemma 2.16 (c)

- Show $\mathbb{R}$ is uncountable.
- By (b),
  $$F = \{f \mid f : \mathbb{N} \to \{0,1\}\}$$
  is uncountable.
- A first idea is to define a function
  $$\begin{array}{rcl} f_0 & : & F \to \mathbb{R} \ , \\ f_0(g) & = & (0.g(0)g(1)g(2)\cdots)_2 \end{array}$$
  Here the right hand side is a number in binary format.
- If f_0 were injective, then by F uncountable we could conclude $\mathbb{R}$ is uncountable.

Proof of Lemma 2.16 (c)

Show $\mathbb{R}$ is uncountable.

- The problem is that
  $$(0.a_0a_1\cdots a_k01111\cdots)_2 \text{ and } (0.a_0a_1\cdots a_k10000\cdots)_2$$
  denote the same real number, so f_0 is not injective.
- We modify f_0 so that we don't obtain any binary numbers of the form
  $$(0.a_0a_1\cdots a_k01111\cdots)_2 \ .$$

Proof of Lemma 2.16 (c)

- Define instead
  $$\begin{array}{rcl} f & : & F \to \mathbb{R} \ , \\ f(g) & := & (0.g(0)\ 0\ g(1)\ 0\ g(2)\ 0\cdots)_2 \ , \end{array}$$
- So
  $$f(g) = (0.a_0a_1a_2\cdots)_2$$
  where
  $$a_k := \begin{cases} 0 & \text{if } k \text{ is odd,} \\ g(\frac{k}{2}) & \text{otherwise.} \end{cases}$$

Proof of Lemma 2.16 (c)

- If two sequences

$$(b_0, b_1, b_2, \ldots) \text{ and } (c_0, c_1, c_2, \ldots)$$

do not end in

$$1, 1, 1, 1, \ldots \quad ,$$

i.e. are not of the form

$$(d_0, d_1, \ldots, d_l, 1, 1, 1, 1, 1, \ldots) \quad ,$$

then one can easily see that

$$(0.b_0 b_1 \cdots)_2 = (0.c_0 c_1 \cdots)_2 \Leftrightarrow (b_0, b_1, b_2, \ldots) = (c_0, c_1, c_2, \ldots)$$

Proof of Lemma 2.16 (c)

Therefore

$$\begin{array}{cl} & f(g) = f(g') \\ \Leftrightarrow & (0.g(0)\ 0\ g(1)\ 0\ g(2)\ 0 \cdots)_2 = (0.g'(0)\ 0\ g'(1)\ 0\ g'(2)\ 0 \cdots)_2 \\ \Leftrightarrow & (g(0)\ , 0,\ g(1),\ 0,\ g(2),\ 0, \ldots) = (g'(0),\ 0,\ g'(1),\ 0,\ g'(2),\ 0, \ldots) \\ \Leftrightarrow & (g(0),\ g(1),\ g(2),\ \ldots) = (g'(0),\ g'(1),\ g'(2),\ \ldots) \\ \Leftrightarrow & g = g' \quad , \end{array}$$

f is injective.

More Uncountable Sets

Lemma 2.17
If A is infinite, then $\mathcal{P}(A)$ and $\{f$ function $\mid f : A \to \{0, 1\}\}$ are uncountable.

Proof: Exercise (reduce it to Lemma 2.16 (a)).

Countable and Complement

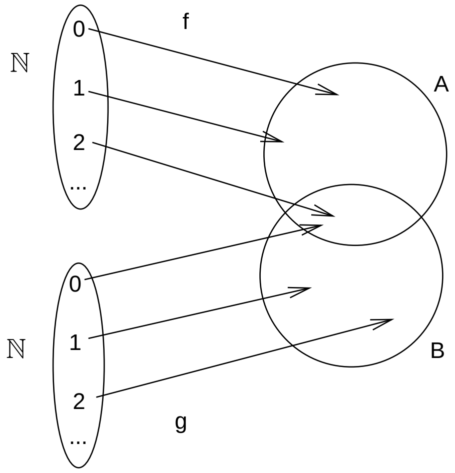
- **Lemma 2.18**
  (a) If A, B are countable, so is $A \cup B$.
  (b) If A is uncountable and B is countable then $A \setminus B$ is uncountable.
- Here $A \setminus B = \{a \in A \mid a \notin B\}$,
  so $A \setminus B$ is A without the elements in B.
- Note that
  - (a) reads: If two sets are small, their union is small as well.
  - (b) reads: If one removes from a big set a small set, then what remains is still big.

Proof of Lemma 2.18 (a)

- To be shown: If A, B are countable, so is $A \cup B$.
- We will use the fact that a set X is countable if and only if it is empty or there exist a surjective function $f : \mathbb{N} \to X$.
- Therefore we need to treat the special cases when A or B are empty.
- Case 1: A is empty.
  Then $A \cup B = B$ which is countable.
- Case 2: B is empty.
  Then $A \cup B = A$ which is countable.

Proof of Lemma 2.18 (a)

Assume $f : \mathbb{N} \to A$, $g : \mathbb{N} \to B$.



Proof of Lemma 2.18 (a)

- Case 3: A, B are not empty.
  - By A, B countable there exist surjective functions

$$f : \mathbb{N} \to A \qquad g : \mathbb{N} \to B$$

  - Define $h : \mathbb{N} \to A \cup B$,

$$h(n) := \begin{cases} f(\frac{n}{2}) & \text{if } n \text{ is even,} \\ g(\frac{n-1}{2}) & \text{if } n \text{ is odd.} \end{cases}$$

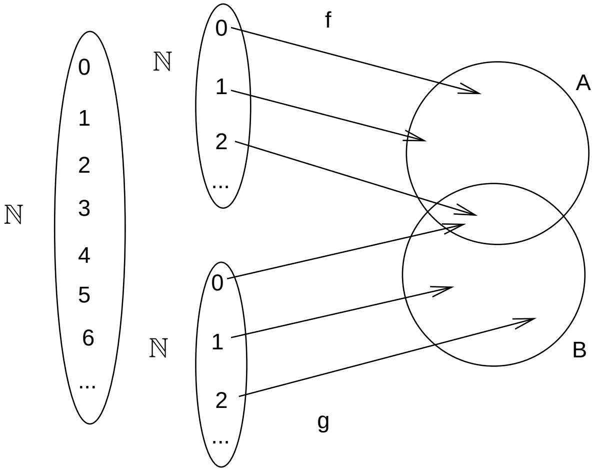
  - So $f(n) = h(2n)$ and $g(n) = h(2n+1)$.
  - Therefore

$$A \cup B = f[\mathbb{N}] \cup g[\mathbb{N}] \subseteq h[\mathbb{N}]$$

  f is surjective.

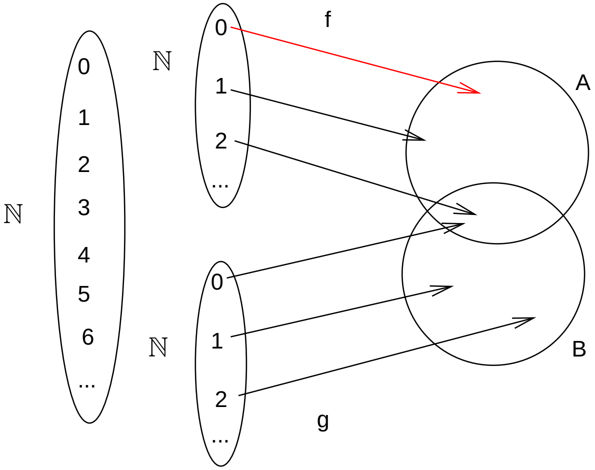
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



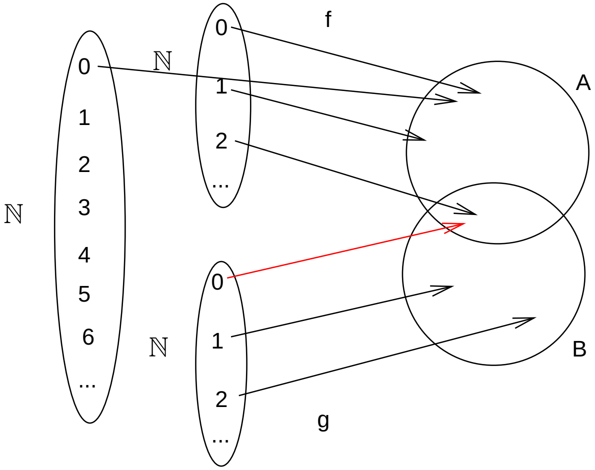
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



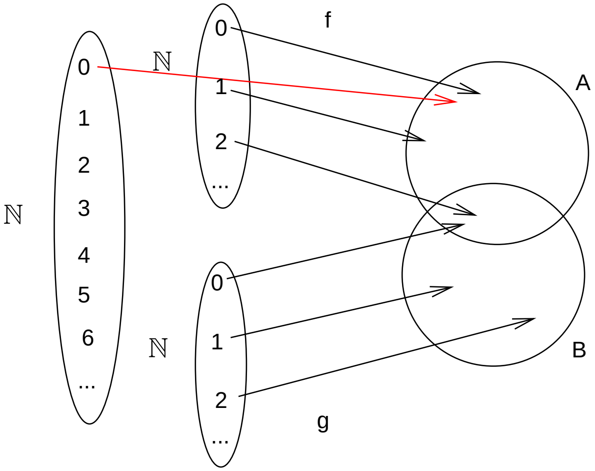
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



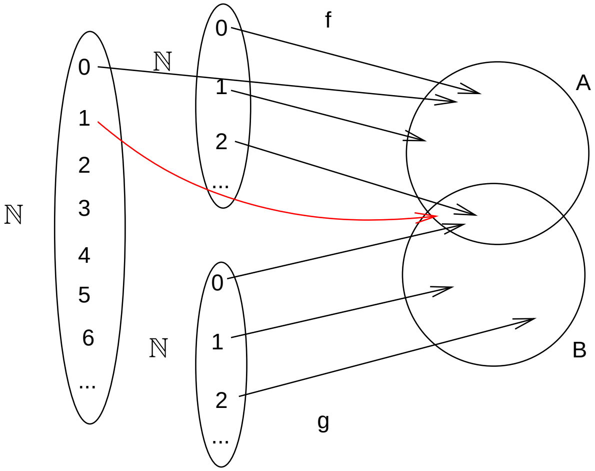
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



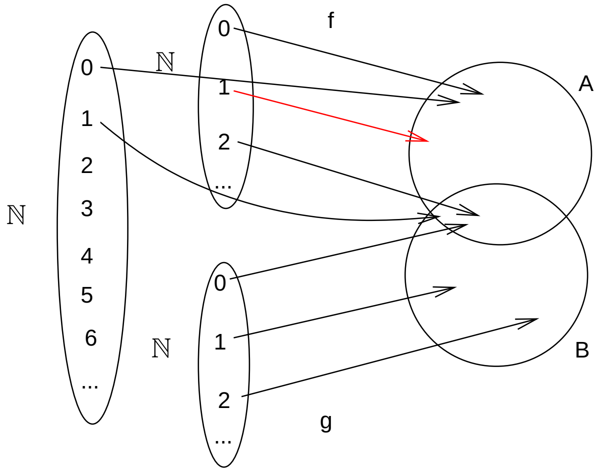
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



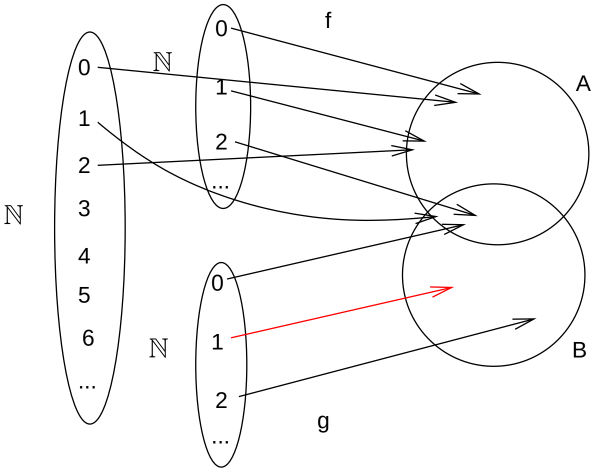
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



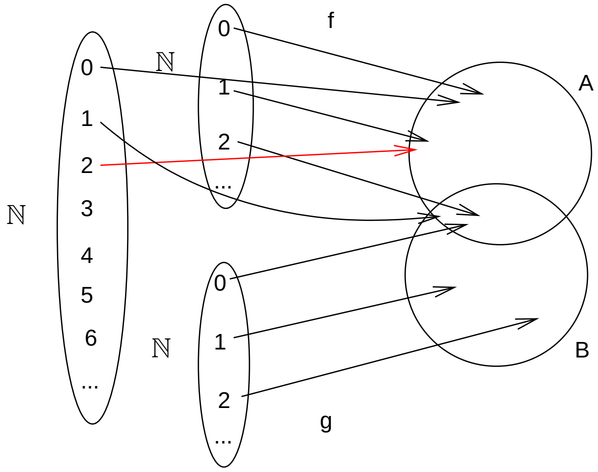
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



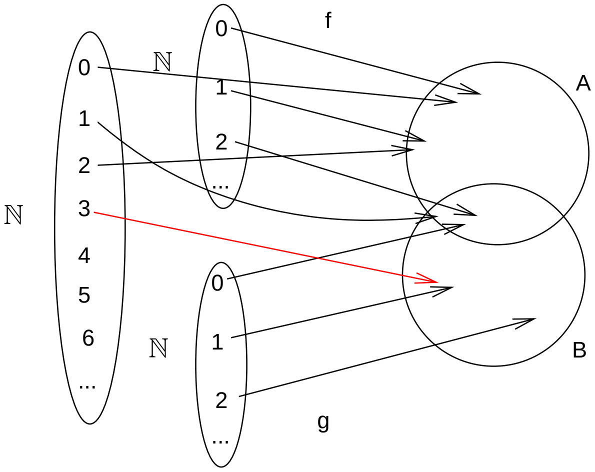
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



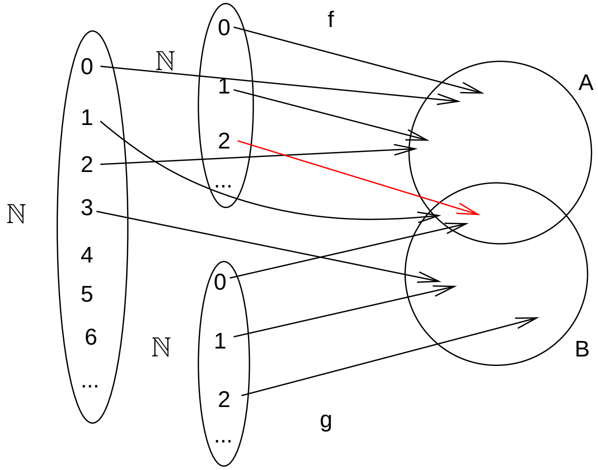
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



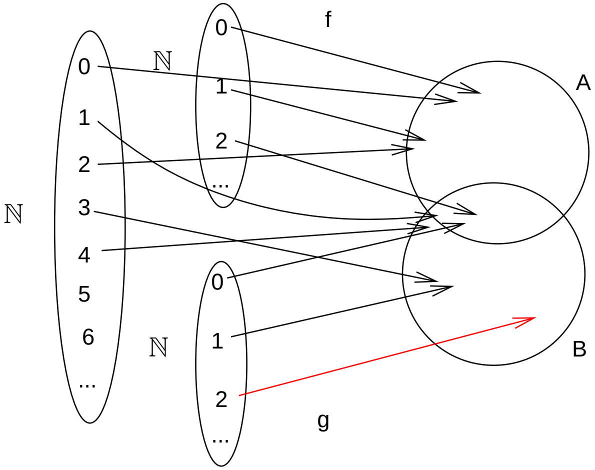
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



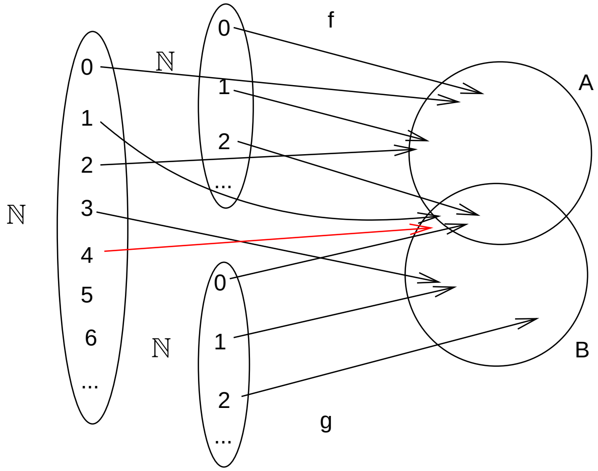
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



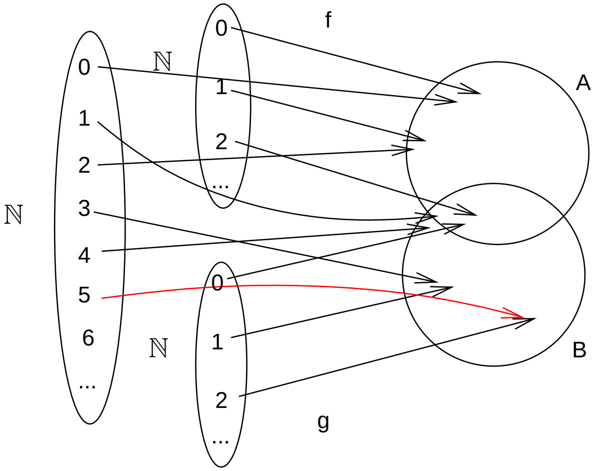
Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



Proof of Lemma 2.18 (a)

$h(2n) = f(n)$, $h(2n+1) = g(n)$:



Jump over the alternative proof.

Alternative Proof of Lemma 2.18 (a)

- To be shown: If A, B are countable, so is $A \cup B$.
- So assume A, B are countable.
- Then there exist (by Lemma 2.11) injective functions

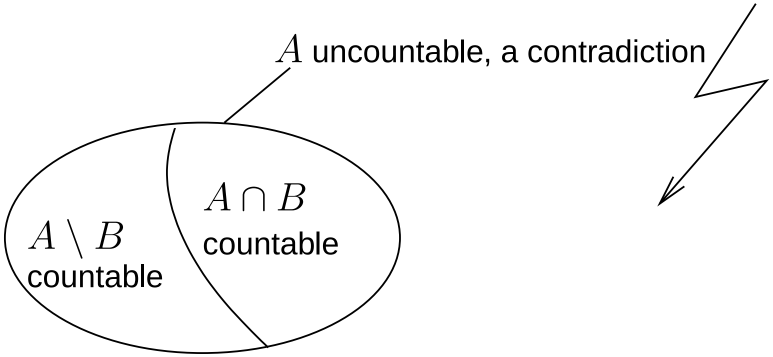
$$f : A \to \mathbb{N} \ , \qquad g : B \to \mathbb{N} \ .$$

- Define

$$\begin{array}{rcl} h & : & A \cup B \to \mathbb{N} \\ h(x) & := & \begin{cases} f(x) \cdot 2 & \text{if } x \in A \\ g(x) \cdot 2 + 1 & \text{if } x \in B \setminus A \end{cases} \end{array}$$

- h is injective.
- Therefore, by Lemma 2.11, $A \cup B$ is countable.

Proof of Lemma 2.18 (b)



Proof of Lemma 2.18 (b)

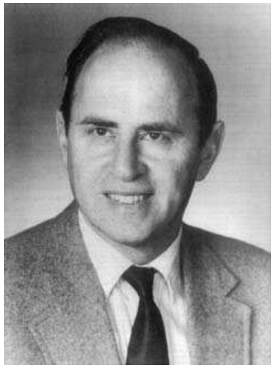
- To be shown:
  If A is uncountable and B is countable, then $A \setminus B$ is uncountable.
- Assume A is uncountable, B is countable and $A \setminus B$ were countable.
- Then $A \cap B$ is countable (since $A \cap B \subseteq B$).
- Therefore $A = (A \setminus B) \cup (A \cap B)$ is countable as well, a contradiction.

Continuum Hypothesis

Remark:

- One can show $\mathcal{P}(\mathbb{N}) \approx \mathbb{R}$.
- Both these sets are uncountable, so they have size bigger than $\mathbb{N}$.
- **Question:** Is there a set B which has size (cardinality) between $\mathbb{N}$ and $\mathbb{R}$?
  - I.e. there are injections $\mathbb{N} \to B$ and $B \to \mathbb{R}$,
  - but neither bijections $\mathbb{N} \to B$ nor $B \to \mathbb{R}$.
- **Continuum Hypothesis**: There exists no such set.
- Continuum Hypothesis is **independent of set theory**, i.e. it is neither provable nor is its negation provable.
  - This was one of the most important open problems in set theory for a long time.

Paul Cohen



Paul Cohen
(1934 – 2007)
Showed 1963 that the continuum hypothesis is independent of set theory.

(d) Reducing Computability to $\mathbb{N}$

- Goal: Reduce computability on some data types A to computability on $\mathbb{N}$.

  A could be for instance the set of strings, of matrices, of trees, of lists of strings, etc.
- If we can do this, then there is no need for a special definition of computability on A, we can concentrate on the notion of computability on $\mathbb{N}$.
- We can reduce computabiliy on A to computability on $\mathbb{N}$, if we have two intuitively computable functions
  - $\mathrm{encode}_A : A \rightarrow \mathbb{N}$,
  - $\mathrm{decode}_A : \mathbb{N} \rightarrow A$.

Reduction of Computability to $\mathbb{N}$

- $\mathrm{encode}_A : A \rightarrow \mathbb{N}$, $\mathrm{decode}_A : \mathbb{N} \rightarrow A$.

- Assume we have such functions encode_A, decode_A, encode_B, decode_B for A and B.
- Then from an intuitively computable $f : A \rightarrow B$ we can obtain an intuitively computable function $\widetilde{f} := \mathrm{encode}_B \circ f \circ \mathrm{decode}_A : \mathbb{N} \rightarrow \mathbb{N}$:

$$\begin{tikzcd}
A \arrow[r, "f"] & B \arrow[d, "\mathrm{encode}_B"] \\
\mathbb{N} \arrow[u, "\mathrm{decode}_A"] \arrow[r, "\widetilde{f}"] & \mathbb{N}
\end{tikzcd}$$

Reduction of Computability to $\mathbb{N}$

- Furthermore from a computable $g : \mathbb{N} \rightarrow \mathbb{N}$ we can obtain an intuitively computable function $\widehat{g} := \mathrm{decode}_B \circ g \circ \mathrm{encode}_A : A \rightarrow B$:

$$\begin{tikzcd}
A \arrow[r, "\widehat{g}"] \arrow[d, "\mathrm{encode}_A"'] & B \\
\mathbb{N} \arrow[r, "g"] & \mathbb{N} \arrow[u, "\mathrm{decode}_B"']
\end{tikzcd}$$

Reduction of Computability to $\mathbb{N}$

- We would like to take the computable functions $g : \mathbb{N} \to \mathbb{N}$ as representations of **all** computable functions $f : A \to B$.
  - In the sense that f represents the function $\widehat{g} : A \to B$.
- This is possible if for any intuitively computable $f : A \to B$ we find a $g : \mathbb{N} \to \mathbb{N}$ s.t. $\widehat{g} = f$.
- We want to use $g = \widetilde{f}$, which is computable, if f is computable.
- But then we need $\widehat{\widetilde{f}} = f$.

Reduction of Computability to $\mathbb{N}$

$\widetilde{f} = \mathrm{encode}_B \circ f \circ \mathrm{decode}_A : \mathbb{N} \to \mathbb{N}$,
$\widehat{g} = \mathrm{decode}_B \circ g \circ \mathrm{encode}_A : A \to B$,
want $\widehat{\widetilde{f}} = f$.

- In order to obtain $\widehat{\widetilde{f}} = f$, we need

$$\begin{aligned} \widehat{\widetilde{f}} &= \mathrm{decode}_B \circ \widetilde{f} \circ \mathrm{encode}_A \\ &= \mathrm{decode}_B \circ \mathrm{encode}_B \circ f \circ \mathrm{decode}_A \circ \mathrm{encode}_A \\ &\overset{!}{=} f \end{aligned}$$

($\overset{!}{=}$ is the equality we need, whereas the other equalities follow by the definition).

Reduction of Computability to $\mathbb{N}$

$\mathrm{decode}_B \circ \mathrm{encode}_B \circ f \circ \mathrm{decode}_A \circ \mathrm{encode}_A \quad \overset{!}{=} \quad f$

- This is fulfilled if we have

$$\begin{aligned} \mathrm{decode}_A \circ \mathrm{encode}_A &= \mathrm{id}_A \\ \mathrm{decode}_B \circ \mathrm{encode}_B &= \mathrm{id}_B \end{aligned}$$

  where id_A is the identity on A, i.e. $\lambda x.x$ similarly for id_B.
- This means that

$$\begin{aligned} &\forall x \in A.\mathrm{decode}_A(\mathrm{encode}_A(x)) = x \\ &\forall x \in B.\mathrm{decode}_B(\mathrm{encode}_B(x)) = x \end{aligned}$$

Reduction of Computability to $\mathbb{N}$

$\forall x \in A.\mathrm{decode}_A(\mathrm{encode}_A(x)) = x$
$\forall x \in B.\mathrm{decode}_B(\mathrm{encode}_B(x)) = x$

- This is a natural condition: If we encode an element of A, and then decode it, we obtain the original element of A back, similarly for B.
  - Note that relationship to cryptography: if we encrypt a message and then decrypt it, we should obtain the original message.

Reduction of Computability to $\mathbb{N}$

- Note that we don't need

$$\mathrm{encode}_A(\mathrm{decode}_A(x)) = x$$

  - Such a condition would mean: every element $n \in \mathbb{N}$ is a code for an element of A (namely $\mathrm{decode}_A(n)$).
  - In cryptography this means: not every element of the datatype of codes is actually an encrypted message.

Computable Encodings

Informal Definition
A data type A has a **computable encoding into $\mathbb{N}$**, *if there exist in an intuitive sense computable functions*

$$\mathrm{encode}_A : A \to \mathbb{N} \quad , \quad \textit{and } \mathrm{decode}_A : \mathbb{N} \to A$$

such that for all $a \in A$ we have

$$\mathrm{decode}_A(\mathrm{encode}_A(a)) = a$$

Computable Encodings

$\mathrm{decode}_A(\mathrm{encode}_A(a)) = a$

- Note that by the above we obtain encode_A is injective.
  - In general we have for two functions $f : B \to C$, $g : C \to D$ that if $g \circ f$ is injective, then f is injective as well.
- Therefore if A has a computable encoding into $\mathbb{N}$, then there exists an injection $\mathrm{encode}_A : A \to \mathbb{N}$, therefore A is countable.

Extension of the Encoding

- We want to show that we have computable encodings of more complex data types into $\mathbb{N}$.
- Assume A and B have computable encodings into $\mathbb{N}$.
- Then we will show that the same applies to
  - $A \times B$, the product of A and B,
  - A^k, the set of k-tuples of A,
  - A^*, the set of lists (or sequences) of elements of A.
- The proof will show as well that if A, B are countable, so are

$$A \times B \quad , \quad A^k, \quad A^* \quad .$$

(e) Encod. of Data Types into $\mathbb{N}$

- In order to show that $A \times B$, A^k, A^* have computable encodings into $\mathbb{N}$, if A, B have, it suffices to show that

$$\mathbb{N} \times \mathbb{N} \quad , \mathbb{N}^n \quad , \mathbb{N}^* \quad ,$$

have computable encodings into $\mathbb{N}$.

- Note that $\mathbb{N}^2 = \mathbb{N} \times \mathbb{N}$.

Reduction to $\mathbb{N}$

- In order to see this assume we had already shown that

$$\mathbb{N}^n \quad , \mathbb{N}^* \quad ,$$

have computable encodings, so we have computable injections

$$\begin{aligned} \mathrm{encode}_{\mathbb{N}^n} &: \mathbb{N}^n \to \mathbb{N} \quad , \\ \mathrm{encode}_{\mathbb{N}^*} &: \mathbb{N}^* \to \mathbb{N} \quad . \end{aligned}$$

with corresonding computable decoding functions.

- Assume A, B have computable with encodings

$$\begin{aligned} \mathrm{encode}_A &: A \to \mathbb{N} \quad , \\ \mathrm{encode}_B &: B \to \mathbb{N} \quad . \end{aligned}$$

Reduction to $\mathbb{N}$

- Then we obtain a computable encoding

$$\begin{aligned} \mathrm{encode}_{A\times B} &: (A \times B) \to \mathbb{N} \\ \mathrm{encode}_{A\times B}((a,b)) &= \underbrace{\mathrm{encode}_{\mathbb{N}^2}\underbrace{((\underbrace{\mathrm{encode}_A(a)}_{\in\mathbb{N}}, \underbrace{\mathrm{encode}_B(b))}_{\in\mathbb{N}}}_{\in\mathbb{N}^2}}_{\in\mathbb{N}} \end{aligned}$$

In short

$$\mathrm{encode}_{A\times B}((a,b)) = \mathrm{encode}_{\mathbb{N}^2}((\mathrm{encode}_A(a), \mathrm{encode}_B(b))$$

Exercise: Define $\mathrm{decode}_{A\times B}$, show $\mathrm{decode}_{A\times B}(\mathrm{encode}_{A\times B}(x)) = x$ and verify that $\mathrm{decode}_{A\times B}$ is intuively computable.

Reduction to $\mathbb{N}$

$$\begin{aligned} \mathrm{encode}_{A^k} &: A^k \to \mathbb{N} \\ \mathrm{encode}_{A^k}((a_0,\ldots,a_{k-1})) &= \\ &\underbrace{\mathrm{encode}_{\mathbb{N}^k}\underbrace{((\underbrace{\mathrm{encode}_A(a_0)}_{\in\mathbb{N}}, \underbrace{\mathrm{encode}_A(a_1)}_{\in\mathbb{N}}, \ldots, \underbrace{\mathrm{encode}_A(a_{k-1}))}_{\in\mathbb{N}}}_{\in\mathbb{N}^k}}_{\in\mathbb{N}} \end{aligned}$$

In short

$$\begin{aligned} &\mathrm{encode}_{A^k}((a_0,\ldots,a_{k-1})) = \\ &\quad \mathrm{encode}_{\mathbb{N}^k}(\mathrm{encode}_A(a_0), \mathrm{encode}_A(a_1), \ldots, \mathrm{encode}_A(a_{k-1})) \end{aligned}$$

Exercise: Define decode_{A^k}, show $\mathrm{decode}_{A^k}(\mathrm{encode}_{A^k}(x)) = x$

and verify that decode_{A^k} is intuively computable.

Reduction to $\mathbb{N}$

We obtain a computable encoding

$$\begin{aligned}\mathrm{encode}_{A^*} &: A^* \longrightarrow \mathbb{N}\\ \mathrm{encode}_{A^*}((a_0,\ldots,a_{n-1})) &= \\ &\underbrace{\mathrm{encode}_{\mathbb{N}^*}(\underbrace{(\underbrace{\mathrm{encode}_A(a_0)}_{\in\mathbb{N}}, \underbrace{\mathrm{encode}_A(a_1)}_{\in\mathbb{N}}, \ldots, \underbrace{\mathrm{encode}_A(a_{n-1})}_{\in\mathbb{N}})}_{\in\mathbb{N}^*})}_{\in\mathbb{N}}\end{aligned}$$

In short

$$\begin{aligned}\mathrm{encode}_{A^*}((a_0,\ldots,a_{n-1})) &= \\ &\mathrm{encode}_{\mathbb{N}^*}((\mathrm{encode}_A(a_0),\mathrm{encode}_A(a_1),\ldots,\mathrm{encode}_A(a_{n-1})))\end{aligned}$$

Exercise: Define decode_{A^*}, show $\mathrm{decode}_{A^*}(\mathrm{encode}_{A^*}(x)) = x$ and verify that decode_{A^*} is intuively computable.

Encoding of Pairs

- The first step is to give a computable encoding of $\mathbb{N}^2$ into $\mathbb{N}$.
- In fact our encoding will be a bijection.
- We will define intuitively computable functions

$$\begin{aligned}\pi &: \mathbb{N}^2 \longrightarrow \mathbb{N}\\ \pi_0 &: \mathbb{N} \longrightarrow \mathbb{N}\\ \pi_1 &: \mathbb{N} \longrightarrow \mathbb{N}\end{aligned}$$

s.t. π and

$$\lambda n.(\pi_0(n),\pi_1(n)) : \mathbb{N} \longrightarrow \mathbb{N}^2$$

are inverse to each other.

Encoding of Pairs

$$\begin{aligned}\pi &: \mathbb{N}^2 \longrightarrow \mathbb{N}\\ \pi_0 &: \mathbb{N} \longrightarrow \mathbb{N}\\ \pi_1 &: \mathbb{N} \longrightarrow \mathbb{N}\end{aligned}$$

- Therefore we obtain a computable encoding of $\mathbb{N}\times\mathbb{N}$ into $\mathbb{N}$ with

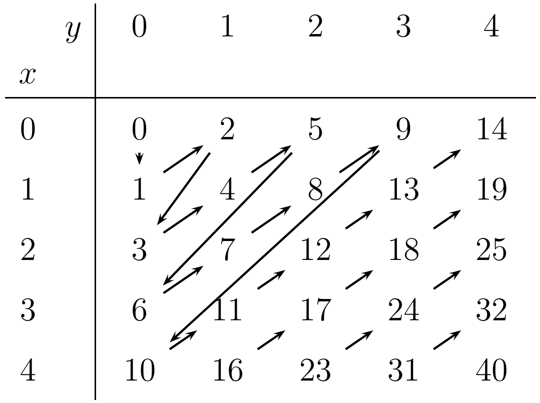
$$\begin{aligned}\mathrm{encode}_{\mathbb{N}\times\mathbb{N}} &:= \pi &&: \mathbb{N}^2 \longrightarrow \mathbb{N}\\ \mathrm{decode}_{\mathbb{N}\times\mathbb{N}} &:= \lambda x.(\pi_0(x),\pi_1(x)) &&: \mathbb{N} \longrightarrow \mathbb{N}^2\end{aligned}$$

Encoding of Pairs

- π will be called the pairing function and π_i the projection functions or short projections.
  π is a computable encoding of $\mathbb{N}^2$ into $\mathbb{N}$.

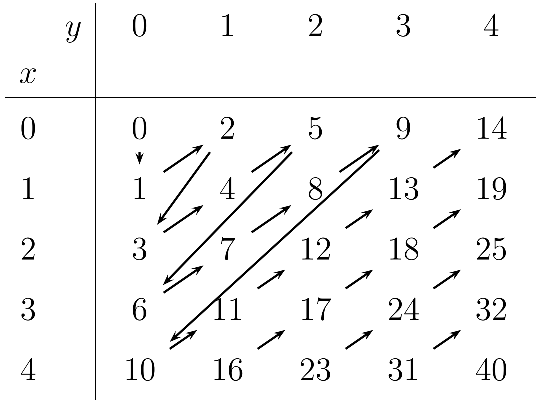
Definition of π

Pairs of natural numbers can be enumerated in the following way:



y x	0	1	2	3	4
0	0	2	5	9	14
1	1	4	8	13	19
2	3	7	12	18	25
3	6	11	17	24	32
4	10	16	23	31	40

Definition of π



y x	0	1	2	3	4
0	0	2	5	9	14
1	1	4	8	13	19
2	3	7	12	18	25
3	6	11	17	24	32
4	10	16	23	31	40

$\pi(0,0) = 0$, $\pi(1,0) = 1$, $\pi(0,1) = 2$,

$\pi(2,0) = 3$, $\pi(1,1) = 4$, $\pi(0,2) = 5$, etc.

Attempt which fails

Note, that the following naïve attempt to enumerate the pairs, fails:

y x	0		1		2		3		
0	$\pi(0,0)$	$\rightarrow$	$\pi(0,1)$	$\rightarrow$	$\pi(0,2)$	$\rightarrow$	$\pi(0,3)$	$\rightarrow$	$\cdots$
1		$\rightarrow$		$\rightarrow$		$\rightarrow$		$\rightarrow$	$\cdots$
2		$\rightarrow$		$\rightarrow$		$\rightarrow$		$\rightarrow$	$\cdots$
3		$\rightarrow$		$\rightarrow$		$\rightarrow$		$\rightarrow$	$\cdots$
4		$\rightarrow$		$\rightarrow$		$\rightarrow$		$\rightarrow$	$\cdots$

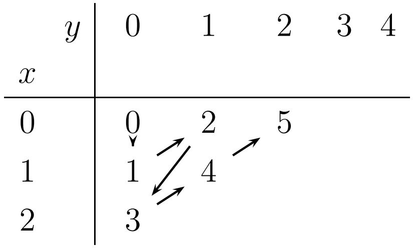
$\pi(0,0) = 0$, $\pi(0,1) = 1$, $\pi(0,2) = 2$, etc.

We never reach the pair $(1,0)$.

Devel. of a Formula for Defining π

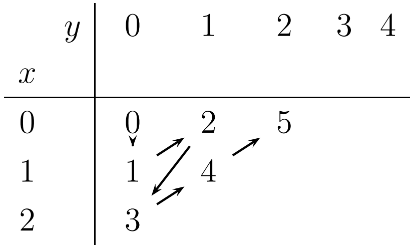
- In the following we are going to develop a mathematical formula for π.
- In the lecture this material was omitted and we give directly the definition of π.
  Jump over Development of π.

Definition of π



y x	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3				

Definition of π

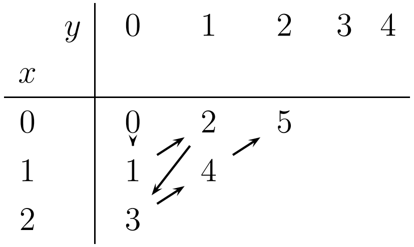


y x	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3				

For the pairs in the diagonal we have the property that $x + y$ is constant.
The first diagonal, consisting of $(0, 0)$ only, is given by

$x + y = 0$.

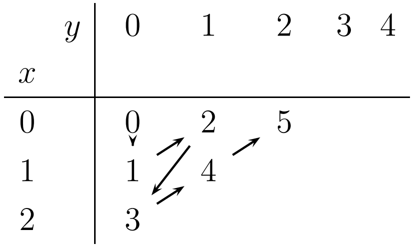
Definition of π



y x	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3				

For the pairs in the diagonal we have the property that $x + y$ is constant.

Definition of π

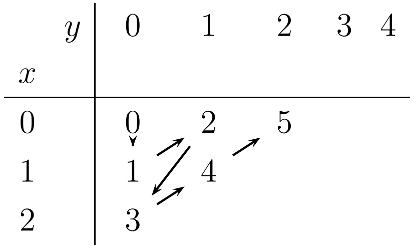


y x	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3				

For the pairs in the diagonal we have the property that $x + y$ is constant.
The second diagonal, consisting of $(1, 0), (0, 1)$, is given by

$x + y = 1$.

Definition of π

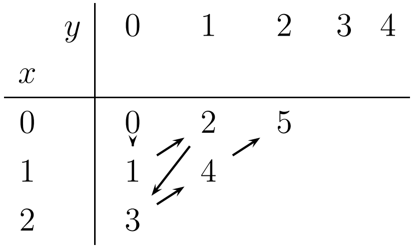


	y	0	1	2	3	4
x						
0		0	2	5		
1		1	4			
2		3				

For the pairs in the diagonal we have the property that $x + y$ is constant.

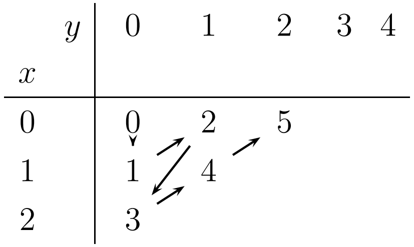
The third diagonal, consisting of $(2, 0), (1, 1), (0, 2)$, is given by $x + y = 2$.

Definition of π



	y	0	1	2	3	4
x						
0		0	2	5		
1		1	4			
2		3				

Definition of π



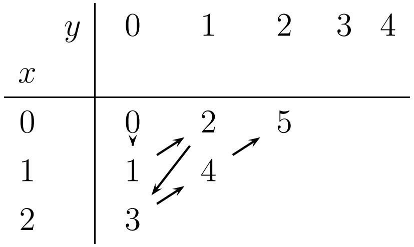
	y	0	1	2	3	4
x						
0		0	2	5		
1		1	4			
2		3				

For the pairs in the diagonal we have the property that $x + y$ is constant.

The third diagonal, consisting of $(2, 0), (1, 1), (0, 2)$, is given by $x + y = 2$.

Etc.

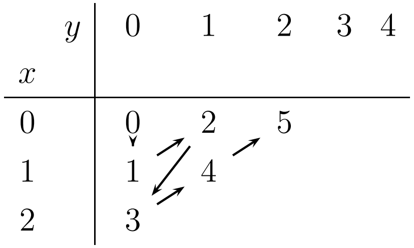
Definition of π



	y	0	1	2	3	4
x						
0		0	2	5		
1		1	4			
2		3				

If we look in the original approach at the diagonals we see that following:

Definition of π

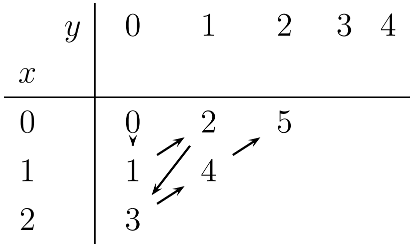


y / x	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3				

If we look in the original approach at the diagonals we see that following:

- The diagonal given by $x + y = n$, consists of $n + 1$ pairs:

Definition of π

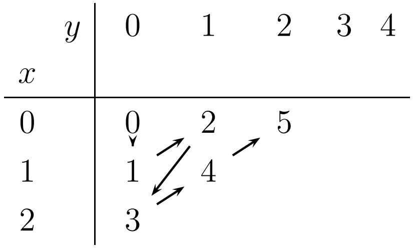


y / x	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3				

If we look in the original approach at the diagonals we see that following:

- The diagonal given by $x + y = n$, consists of $n + 1$ pairs:
  - The first diagonal, given by $x + y = 0$, consists of $(0, 0)$ only, i.e. of 1 pair.

Definition of π

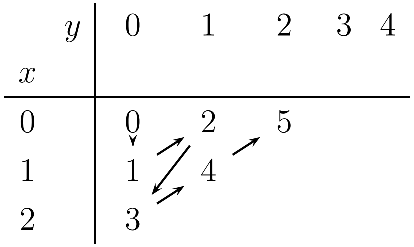


y / x	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3				

If we look in the original approach at the diagonals we see that following:

- The diagonal given by $x + y = n$, consists of $n + 1$ pairs:
  - The second diagonal, given by $x + y = 1$, consists of $(1, 0), (0, 1)$, i.e. of 2 pairs.

Definition of π

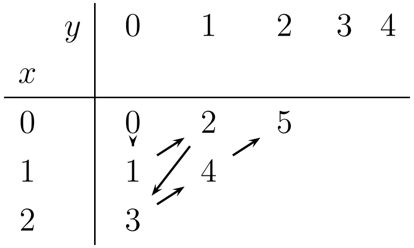


y / x	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3				

If we look in the original approach at the diagonals we see that following:

- The diagonal given by $x + y = n$, consists of $n + 1$ pairs:
  - The third diagonal, given by $x + y = 2$, consisting of $(2, 0), (1, 1), (0, 2)$, i.e. of 3 pairs.

Definition of π

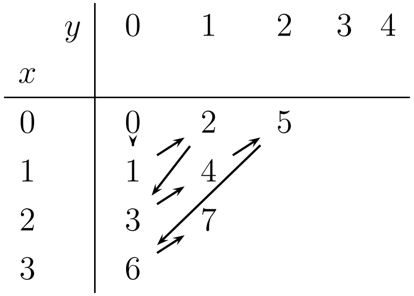


x \ y	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3				

If we look in the original approach at the diagonals we see that following:

- The diagonal given by $x + y = n$, consists of $n + 1$ pairs:
  - The third diagonal, given by $x + y = 2$, consisting of $(2, 0), (1, 1), (0, 2)$, i.e. of 3 pairs.
  - etc.

Definition of π

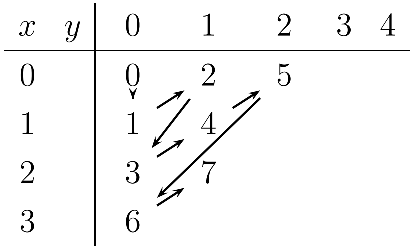


x \ y	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3	7			
3	6				

We count the elements occurring before the pair (x_0, y_0).

- We have to count all elements of the previous diagonals. These are those given by $x + y = n$ for $n < x_0 + y_0$.
  - In the above example for the pair $(2, 1)$, these are the diagonals given by $x + y = 0$, $x + y = 1$, $x + y = 2$.

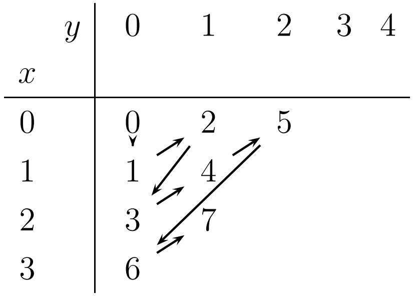
Definition of π



x y	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3	7			
3	6				

- The diagonal, given by $x + y = n$, has $n + 1$ elements, so in total we have $\sum_{i=0}^{x+y-1}(i + 1) = 1 + 2 + \cdots + (x + y) = \sum_{i=1}^{x+y} i$ elements in those diagonals.
- A often used formula says $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. Therefore, the above is $\frac{(x+y)(x+y+1)}{2}$.

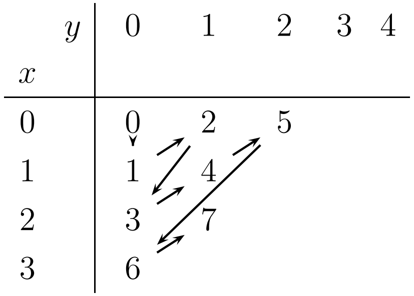
Definition of π



x \ y	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3	7			
3	6				

- Further, we have to count all pairs in the current diagonal, which occur in this ordering before the current one. These are y pairs.
  - Before $(2, 1)$ there is only one pair, namely $(3, 0)$.
  - Before $(3, 0)$ there are 0 pairs.
  - Before $(0, 2)$ there are 2 pairs, namely $(2, 0), (1, 1)$.

Definition of π



$x \backslash y$	0	1	2	3	4
0	0	2	5		
1	1	4			
2	3	7			
3	6				

- Therefore we get that there are in total $\frac{(x+y)(x+y+1)}{2} + y$ pairs before (x, y), therefore the pair (x, y) is the pair number $(\frac{(x+y)(x+y+1)}{2} + y)$ in this order.

Definition of π

Definition 2.19

$$\pi(x,y) := \frac{(x+y)(x+y+1)}{2} + y \qquad (= (\sum_{i=1}^{x+y} i) + y)$$

Exercise: Prove that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

π is Bijective

Lemma 2.20
π *is bijective.*

Omit Proof

Proof of Bijectivity of π

We show π **is injective:**
We prove first that, if $x + y < x' + y'$, then $\pi(x, y) < \pi(x', y')$:

$$\begin{aligned}\pi(x,y) = (\sum_{i=1}^{x+y} i) + y \quad &< \quad (\sum_{i=1}^{x+y} i) + x + y + 1 = \sum_{i=1}^{x+y+1} i \\ &\leq \quad (\sum_{i=1}^{x'+y'} i) + y' = \pi(x', y')\end{aligned}$$

Proof of Bijectivity of π

We show π is injective:
Assume now $\pi(x,y) = \pi(x',y')$ and show $x = x'$ and $y = y'$.
We have by the above

$$x + y = x' + y' \ .$$

Therefore

$$y = \pi(x,y) - (\sum_{i=1}^{x+y} i) = \pi(x',y') - (\sum_{i=1}^{x'+y'} i) = y'$$

and

$$x = (x+y) - y = (x'+y') - y' = x' \ .$$

Proof of Bijectivity of π

We show π **is surjective:**
Assume $n \in \mathbb{N}$.
Show $\pi(x,y) = n$ for some $x, y \in \mathbb{N}$.
The sequence $(\sum_{i=1}^{k'} i)_{k' \in \mathbb{N}}$ is strictly existing.
Therefore there exists a k s.t.

$$a := \sum_{i=1}^{k} i \leq n < \sum_{i=1}^{k+1} i$$

Proof of Bijectivity of π

$n \in \mathbb{N}$
Show $\pi(x,y) = n$ for some x, y

$$a := \sum_{i=1}^{k} i \leq n < \sum_{i=1}^{k+1} i \qquad (*)$$

So, in order to obtain $\pi(x,y) = n$, we need $x + y = k$.
By $y = \pi(x,y) - \sum_{i=1}^{x+y} i$, we need to define $y := n - a$.
By $k = x + y$, we need to define $x := k - y$.
By $(*)$ it follows $0 \leq y < k + 1$,
therefore $x, y \geq 0$. Further,
$\pi(x,y) = (\sum_{i=1}^{x+y} i) + y = (\sum_{i=1}^{k} i) + (n - \sum_{i=1}^{k} i) = n$.

Definition of π_0, π_1

Since π is bijective, we can define π_0, π_1 as follows:

Definition 2.21
Let $\pi_0 : \mathbb{N} \to \mathbb{N}$ *and* $\pi_1 : \mathbb{N} \to \mathbb{N}$ *be s.t.*

$$\pi_0(\pi(x,y)) = x \ , \quad \pi_1(\pi(x,y)) = y \ .$$

π, π_i are Computable

Remark
π, π_0, π_1 are computable in an intuitive sense.

“Proof:”

- π is obviously computable.
- In order to compute π_0, π_1, first observe that $x, y \leq \pi(x, y)$.
  - Follows from $\pi(x, y) = (\sum_{i=1}^{x+y} i) + y$.
- Therefore $\pi_0(n)$, $\pi_1(n)$ can be computed by
  - searching for $x, y \leq n$ s.t. $\pi(x, y) = n$,
  - and then setting $\pi_0(n) = x$, $\pi_1(n) = y$.

Remark 2.22

Remark 2.22
For all $z \in \mathbb{N}$*,*

$$\pi(\pi_0(z), \pi_1(z)) = z \quad .$$

Proof:
Assume $z \in \mathbb{N}$ and show

$$z = \pi(\pi_0(z), \pi_1(z)) \quad .$$

π is surjective, so there exists x, y s.t.

$$\pi(x, y) = z \quad .$$

Then

$$\pi(\pi_0(z), \pi_1(z)) = \pi(\pi_0(\pi(x, y)), \pi_1(\pi(x, y))) = \pi(x, y) = z.$$

Encoding of $\mathbb{N}^k$

- We want to encode $\mathbb{N}^k$ into $\mathbb{N}$.
- $(l, m, n) \in \mathbb{N}^3$ can be encoded as follows
  - First encode (l, m) as $\pi(l, m) \in \mathbb{N}$.
  - Then encode the complete triple as

$$\pi(\pi(l, m), n) \in \mathbb{N} \quad .$$

  - So define

$$\pi^3(l, m, n) := \pi(\pi(l, m), n) \quad .$$

- Similarly $(l, m, n, p) \in \mathbb{N}^4$ can be encoded as follows:

$$\pi^4(l, m, n, p) := \pi(\pi(\pi(l, m), n), p) \quad .$$

Decoding Function

- If $x = \pi^3(l, m, n) = \pi(\pi(l, m), n)$, then we see
  - $l = \pi_0(\pi_0(x))$,
  - $m = \pi_1(\pi_0(x))$,
  - $n = \pi_1(x)$.
- So we define
  - $\pi^3_0(x) = \pi_0(\pi_0(x))$,
  - $\pi^3_1(x) = \pi_1(\pi_0(x))$,
  - $\pi^3_2(x) = \pi_1(x)$.

Decoding Function

- Similarly, if $x = \pi^4(l,m,n,p) = \pi(\pi(\pi(l,m),n),p)$, then we see
  - $l = \pi_0(\pi_0(\pi_0(x)))$,
  - $m = \pi_1(\pi_0(\pi_0(x)))$,
  - $n = \pi_1(\pi_0(x))$.
  - $p = \pi_1(x)$.
- So we define
  - $\pi^4_0(x) = \pi_0(\pi_0(\pi_0(x)))$,
  - $\pi^4_1(x) = \pi_1(\pi_0(\pi_0(x)))$,
  - $\pi^4_2(x) = \pi_1(\pi_0(x))$.
  - $\pi^4_3(x) = \pi_1(x)$.

Formal definition of π^k, π^k_i

- Then π^k and

  $$\lambda x.(\pi^k_0(x),\ldots,\pi^k_{k-1}(x))$$

  are inverse to each other.
- A formal inductive Definition of π^k and π^k_i is as follows: Jump over formal definition of π

Definition for General k

In general one defines for $k \geq 1$

$$\begin{array}{rcl} \pi^k &:& \mathbb{N}^k \to \mathbb{N} \quad , \\ \pi^k(x_0,\ldots,x_{k-1}) &:=& \pi(\cdots\pi(\pi(x_0,x_1),x_2)\cdots x_{k-1}) \quad , \end{array}$$

and for $i < k$

$$\begin{array}{rcl} \pi^k_i &:& \mathbb{N} \to \mathbb{N} \quad , \\ \pi^k_0(x) &:=& \underbrace{\pi_0(\cdots\pi_0(}_{k-1 \text{ times}}\ x)\cdots) \quad , \end{array}$$

and for $0 < i < k$,

$$\pi^k_i(x) \ := \ \pi_1(\ \underbrace{\pi_0(\pi_0(\cdots\pi_0(}_{k-i-1 \text{ times}}\ x)\cdots))) \quad .$$

Definition 2.23 of π^k, π^k_i

(a) *We define by induction on k for $k \in \mathbb{N}$, $k \geq 1$*

$$\begin{array}{rrcl} & \pi^k &:& \mathbb{N}^k \to \mathbb{N} \\ & \pi^1(x) &:=& x \\ \textit{For } k > 0 & \pi^{k+1}(x_0,\ldots,x_k) &:=& \pi(\pi^k(x_0,\ldots,x_{k-1}),x_k) \end{array}$$

(b) *We define by induction on k for $i, k \in \mathbb{N}$ s.t. $1 \leq k$, $0 \leq i < k$*

$$\begin{array}{rcl} \pi^k_i &:& \mathbb{N} \to \mathbb{N} \\ \pi^1_0(x) &:=& x \\ \pi^{k+1}_i(x) &:=& \pi^k_i(\pi_0(x)) \textit{ for } i < k \\ \pi^{k+1}_k(x) &:=& \pi_1(x) \end{array}$$

Omit Examples.

Examples

- $\pi^2(x,y) = \pi(\pi^1(x),y) = \pi(x,y)$.
- $\pi^3(x,y,z) = \pi(\pi^2(x,y),z) = \pi(\pi(x,y),z)$.
- $\pi^4(x,y,z,u) = \pi(\pi^3(x,y,z),u) = \pi(\pi(\pi(x,y),z),u)$.
- $\pi^4_0(u) = \pi^3_0(\pi_0(u)) = \pi^2_0(\pi_0(\pi_0(u))) = \pi^1_0(\pi_0(\pi_0(\pi_0(u)))) = \pi_0(\pi_0(\pi_0(u)))$.
- $\pi^4_2(u) = \pi^3_2(\pi_0(u)) = \pi_1(\pi_0(u))$.

Proof

Induction on k.
Base case $k = 0$:
Proof of (a):
Let $(x_0) \in \mathbb{N}^1$.
Then $\pi^1_0(\pi^1(x_0)) = x_0$.
Proof of (b):
Let $x \in \mathbb{N}$.
Then $\pi^1(\pi^1_0(x)) = x$.

Lemma 2.24

(a) *For* $(x_0,\ldots,x_{k-1}) \in \mathbb{N}^k$*,* $i < k$*,* $x_i = \pi^k_i(\pi^k(x_0,\ldots,x_{k-1}))$.

(b) *For* $x \in \mathbb{N}$*,* $x = \pi^k(\pi^k_0(x),\ldots,\pi^k_{k-1}(x))$.

(Omit Proof)

Proof of Lemma 2.24

Induction step $k \to k+1$:
Assume the assertion has been shown for k.
Proof of (a):
Let $(x_0,\ldots,x_k) \in \mathbb{N}^{k+1}$.
Then

$$\begin{aligned} \text{for } i < k \quad & \pi^{k+1}_i(\pi^{k+1}(x_0,\ldots,x_k)) \\ = \quad & \pi^k_i(\pi_0(\pi(\pi^k(x_0,\ldots,x_{k-1}),x_k))) \\ = \quad & \pi^k_i(\pi^k(x_0,\ldots,x_{k-1})) \\ \overset{\text{IH}}{=} \quad & x_i \end{aligned}$$

$$\begin{aligned} \text{and } & \pi^{k+1}_k(\pi^{k+1}(x_0,\ldots,x_k)) \\ = \quad & \pi_1(\pi(\pi^k(x_0,\ldots,x_{k-1}),x_k)) \\ = \quad & x_k \end{aligned}$$

Proof of Lemma 2.24

Induction step $k \to k+1$:
Assume the assertion has been shown for k.

Proof of (b):
Let $x \in \mathbb{N}$.

$$\begin{aligned} & \pi^{k+1}(\pi_0^{k+1}(x),\ldots,\pi_k^{k+1}(x)) \\ = \quad & \pi(\pi^k(\pi_0^{k+1}(x),\ldots,\pi_{k-1}^{k+1}(x)),\pi_k^{k+1}(x)) \\ = \quad & \pi(\pi^k(\pi_0^k(\pi_0(x)),\ldots,\pi_{k-1}^k(\pi_0(x))),\pi_1(x)) \\ \overset{\text{IH}}{=} \quad & \pi(\pi_0(x),\pi_1(x)) \\ \overset{\text{Rem. 2.22}}{=} \quad & x \end{aligned}$$

Encoding of $\mathbb{N}^*$

- Note that each $n \in \mathbb{N}$ is a code for elements of $\mathbb{N}^k$ for every k.
  - So if we encoded $(n_0,\ldots,n_{k-1})$ as $\pi^k(n_0,\ldots,n_{k-1})$ we couldn't determine the length k of the original sequence from the code.
- So we need to add the length to the code for $(n_0,\ldots,n_{k-1})$ (considered as an element of $\mathbb{N}^*$).
- Therefore encode a sequence $(n_0,\ldots,n_{k-1}) \in \mathbb{N}^*$ for $k > 0$ as

$$\pi(k-1,\pi^k(n_0,\ldots,n_{k-1})) \ .$$

- In order to distinguish it from code of $()$, add 1 to it.
- In total we obtain a bijection.

Encoding of $\mathbb{N}^*$

- We want to define an encoding $\mathrm{encode}_{\mathbb{N}^*} : \mathbb{N}^* \to \mathbb{N}$ (which will be a bijection).
- $\mathbb{N}^* = \mathbb{N}^0 \cup \bigcup_{k\geq 1} \mathbb{N}^k$.
- $\mathbb{N}^0 = \{()\}$,
  We can encode $()$ as 0.
- Encoding of $\bigcup_{k\geq 1} \mathbb{N}^k$:
  - We have an encoding

$$\pi^k : \mathbb{N}^k \to \mathbb{N} \ .$$

Definition 2.25 of $\langle\rangle$, lh, $(x)_i$

(a) Define for $x \in \mathbb{N}^*$, $\langle x \rangle : \mathbb{N}$ as follows:

$$\langle\rangle \quad := \quad \langle ()\rangle := 0 \ ,$$

for $k > 0$

$$\begin{aligned} \langle n_0,\ldots,n_{k-1}\rangle &:= \langle (n_0,\ldots,n_{k-1})\rangle \\ &:= 1 + \pi(k-1,\pi^k(n_0,\ldots,n_{k-1})) \end{aligned}$$

(b) Define for $x \in \mathbb{N}$, the **length** lh$(x) \in \mathbb{N}$ as follows:

$$\begin{aligned} \mathrm{lh} \quad &: \quad \mathbb{N} \to \mathbb{N} \ , \\ \mathrm{lh}(0) \quad &:= \quad 0 \ , \\ \mathrm{lh}(x) \quad &:= \quad \pi_0(x-1) + 1 \text{ if } x > 0 \ . \end{aligned}$$

Definition 2.25 of $\langle\rangle$, lh, $(x)_i$

(c) We define for $x \in \mathbb{N}$ and $i < \text{lh}(x)$, the ith component

$$(x)_i \in \mathbb{N}$$

of a code x for a sequence as follows:

$$(x)_i := \pi_i^{\text{lh}(x)}(\pi_1(x-1)) \ .$$

For $\text{lh}(x) \leq i$, let

$$(x)_i := 0 \ .$$

$(x_0,\ldots,x_{k-1})$ vs. $\langle x_0,\ldots,x_{k-1}\rangle$

Remark:

(a) Note that $(x_0,\ldots,x_{k-1})$ is a tuple, which is an element of $\mathbb{N}^k$, whereas $\langle x_0,\ldots,x_{k-1}\rangle$ is the code for this tuple, which is an element of $\mathbb{N}$.

(b) Especially $() \in \mathbb{N}^0$ is the empty tuple, whereas $\langle\rangle = 0 \in \mathbb{N}$ is the code for the empty tuple.

Remark

- $\text{lh}(x)$, $(x)_i$ are defined in such a way that Lemma 2.26 (a), (b) given below hold.
- This shows that lh, $(x)_i$ together form the inverse of the forming of $\langle x_0,\ldots,x_{k-1}\rangle$.

Lemma 2.26

Lemma 2.26

(a) $\text{lh}(\langle\rangle) = 0$, $\text{lh}(\langle n_0,\ldots,n_k\rangle) = k+1$.

(b) *For* $i \leq k$, $(\langle n_0,\ldots,n_k\rangle)_i = n_i$.

(c) *For* $x \in \mathbb{N}$, $x = \langle (x)_0,\ldots,(x)_{\text{lh}(x)-1}\rangle$.

Remark

If we define

$$\begin{aligned}&\langle\rangle^{-1} : \mathbb{N} \to \mathbb{N}^* \\ &\langle\rangle^{-1}(x) = ((x)_0, \ldots, (x)_{\mathrm{lh}(x)-1})\end{aligned}$$

Then we have by Lemma 2.26

$$\langle\rangle^{-1}(\langle x_0, \ldots, x_{n-1}\rangle) = (x_0, \ldots, x_{n-1})$$

so $\langle\rangle^{-1}$ is the inverse of $\vec{x} \mapsto \langle\vec{x}\rangle$.

(Omit Proof of Lemma 2.26)

Proof of Lemma 2.26 (a)

Proof of (a):

Show: $\mathrm{lh}(\langle\rangle) = 0$:
$\mathrm{lh}(\langle\rangle) = \mathrm{lh}(0) = 0$.

Show: $\mathrm{lh}(\langle n_0, \ldots, n_k\rangle) = k + 1$:

$$\begin{aligned}\mathrm{lh}(\langle n_0, \ldots, n_k\rangle) &= \pi_0(\langle n_0, \ldots, n_k\rangle - 1) + 1 \\ &= \pi_0(\pi(k, \cdots) + 1 - 1) + 1 \\ &= k + 1\end{aligned}$$

Proof of Lemma 2.26 (b)

Proof of (b):
Show $(\langle n_0, \ldots, n_k\rangle)_i = n_i$.
$\mathrm{lh}(\langle n_0, \ldots, n_k\rangle) = k + 1$.
Therefore

$$\begin{aligned}& (\langle n_0, \ldots, n_k\rangle)_i \\ = \;& \pi_i^{k+1}(\pi_1(\langle n_0, \ldots, n_k\rangle - 1)) \\ = \;& \pi_i^{k+1}(\pi_1(1 + \pi(k, \pi^{k+1}(n_0, \ldots, n_k)) - 1)) \\ = \;& \pi_i^{k+1}(\pi^{k+1}(n_0, \ldots, n_k)) \\ \overset{\text{Lem 2.24 (a)}}{=} \;& n_i\end{aligned}$$

Proof of Lemma 2.26 (c)

Proof of (c):
Show $x = \langle (x)_0, \ldots, (x)_{\mathrm{lh}(x)-1}\rangle$.

Case $x = 0$.
$\mathrm{lh}(x) = 0$. Therefore $\langle (x)_0, \ldots, (x)_{\mathrm{lh}(x)-1}\rangle = \langle\rangle = 0 = x$.

Case $x > 0$.
Let $x - 1 = \pi(l, y)$.
Then $\mathrm{lh}(x) = l + 1$, $(x)_i = \pi_i^{l+1}(y)$ and therefore

$$\begin{aligned}& \langle (x)_0, \ldots, (x)_{\mathrm{lh}(x)-1}\rangle \\ = \;& \langle \pi_0^{l+1}(y), \ldots, \pi_l^{l+1}(y)\rangle \\ = \;& \pi(l, \pi^{l+1}(\pi_0^{l+1}(y), \ldots, \pi_l^{l+1}(y))) + 1 \\ \overset{\text{Lem 2.24 (b)}}{=} \;& \pi(l, y) + 1 \\ = \;& x\end{aligned}$$

Encoding of Finite Sets, Strings

Informal Lemma

If A is a finite non-empty set, then A and A^* have computables encoding into $\mathbb{N}$.

Proof of the Informal Lemma

- Assume

$$A = \{a_0, \ldots, a_n\}$$

where $a_i \neq a_j$ for $i \neq j$, $n \geq 0$.

- Define

$$\begin{array}{rcl} \mathrm{encode}_A & : & A \to \mathbb{N} \\ \mathrm{encode}_A(a_i) & = & i \ . \end{array}$$

Define

$$\begin{array}{rcl} \mathrm{decode}_A & : & \mathbb{N} \to A \\ \mathrm{decode}_A(i) & := & a_i \text{ if } i \leq n \\ \mathrm{decode}_A(i) & := & a_0 \text{ if } i > n. \end{array}$$

Proof of the Informal Lemma

- encode_A and decode_A are in an intuitive sense computable, and

$$\mathrm{decode}_A(\mathrm{encode}_A(a)) = a$$

- Therefore A has a computable encoding into $\mathbb{N}$,
- Therefore A^* has as well a computable encoding into $\mathbb{N}$.

Remark: One easily sees that the encoding obtained by this proof is

$$\begin{array}{rcl} \mathrm{encode}_{A^*} & : & A^* \to \mathbb{N} \ , \\ \mathrm{encode}_{A^*}(a_0, \ldots, a_n) & = & \langle \mathrm{encode}_A(a_0), \ldots, \mathrm{encode}_A(a_n) \rangle \end{array}$$

Theorem 2.27

Theorem 2.27

(a) $\mathbb{N}^k$ and $\mathbb{N}^*$ are countable.

(b) If A is countable, so are A^k, A^*.

(c) If A, B are countable, so is $A \times B$.

(d) If A_n are countable sets for $n \in \mathbb{N}$, so is $\bigcup_{n \in \mathbb{N}} A_n$.

(e) $\mathbb{Q}$, the set of rational numbers, is countable.

Proof of Theorem 2.27 (a)

- $\mathbb{N}^0 = \{()\}$ is finite therefore countable.
- For $k > 0$

$$\pi^k : \mathbb{N}^k \to \mathbb{N}$$

  is a bijection.
- The function

$$\lambda x.\langle x\rangle : \mathbb{N}^* \to \mathbb{N}$$

  is a bijection.

Proof of Theorem 2.27 (b)

- To be shown: If A is countable, so are A^k, A^*.
- Assume A is countable.
- We show first that A^* is countable:
- There exists $\mathrm{encode}_A : A \to \mathbb{N}$, encode_A injective.
- Define

$$\begin{array}{rcl} f & : & A^* \to \mathbb{N}^* \ , \\ f(a_0, \ldots, a_{k-1}) & := & (\mathrm{encode}_A(a_0), \ldots, \mathrm{encode}_A(a_{k-1})) \end{array}$$

- f is injective as well, $\mathbb{N}^*$ is countable, so by Corollary 2.13 A^* is countable.
- $A^k \subseteq A^*$, so A^k is countable as well.

Proof of Theorem 2.27 (c)

- Assume A, B countable.
- Then there exist injections

$$\begin{array}{rcl} \mathrm{encode}_A & : & A \to \mathbb{N} \\ \mathrm{encode}_B & : & B \to \mathbb{N} \end{array}$$

- Define

$$\begin{array}{rcl} f & : & (A \times B) \to \mathbb{N}^2 \ , \\ f(a, b) & := & (\mathrm{encode}_A(a), \mathrm{encode}_B(b)) \end{array}$$

- f is injective, $\mathbb{N}^2$ is countable, so $A \times B$ is countable as well.

Proof of Theorem 2.27 (d)

- Assume A_n are countable for $n \in \mathbb{N}$.
- Show

$$A := \bigcup_{n\in\mathbb{N}} A_n$$

  is countable as well.
- If all A_n are empty, so is

$$\bigcup_{n\in\mathbb{N}} A_n$$

  and therefore countable.
- Assume now A_{k_0} is non-empty for some k_0.

Proof of Theorem 2.27 (d)

A_n are countable
Show $\bigcup_{n\in\mathbb{N}} A_n$ is countable.

- By replacing empty A_l by A_{k_0}, we get a sequence of non-empty sets $(A_n)_{n\in\mathbb{N}}$, s.t. their union is the same as A.
- So we can assume without loss of generality $A_n \neq \emptyset$ for all n.
- A_n are countable and non-empty, so there exist $f_n : \mathbb{N} \to A_n$ surjective.

Proof of Theorem 2.27 (d)

$f_n : \mathbb{N} \to A_n$ surjective
Show $\bigcup_{n\in\mathbb{N}} A_n$ is countable.

- Then
$$\begin{array}{rcl} f & : & \mathbb{N}^2 \to \bigcup_{n\in\mathbb{N}} A_n \ , \\ f(n,m) & := & f_n(m) \end{array}$$
is surjective as well.
- $\mathbb{N}^2$ is countable, so by Corollary 2.15 A is countable as well.

Proof of Theorem 2.27 (e)

- To be shown: $\mathbb{Q}$ is countable.
- We have $\mathbb{Z} \times \mathbb{N}$ is countable, since $\mathbb{Z}$ and $\mathbb{N}$ are countable.
- Let
$$A := \{(z,n) \in \mathbb{Z} \times \mathbb{N}, n \neq 0\} \ .$$
- $A \subseteq \mathbb{Z} \times \mathbb{N}$, therefore A is countable as well.
- Define
$$\begin{array}{rcl} g & : & A \to \mathbb{Q} \ , \\ g(z,n) & := & \frac{z}{n} \ . \end{array}$$
- g is surjective, A countable, therefore by Corollary 2.15 $\mathbb{Q}$ is countable as well.

(f) Partial Functions

- A partial function $f : A \overset{\sim}{\to} B$ is the same as a function $f : A \to B$, but $f(a)$ might not be defined for all $a \in A$.
- Key example: function computed by a computer program:
  - Program has some input $a \in A$ and possibly returns some $b \in B$.
    (We assume that program does not refer to global variables).
  - If the program applied to $a \in A$ terminates and returns b, then $f(a)$ is defined and equal to b.
  - If the program applied to $a \in A$ does not terminate, then $f(a)$ is undefined.

Examples of Partial Functions

- Other Examples:
  - $f : \mathbb{R} \overset{\sim}{\to} \mathbb{R}$, $f(x) = \frac{1}{x}$:
    $f(0)$ is undefined.
  - $g : \mathbb{R} \overset{\sim}{\to} \mathbb{R}$, $g(x) = \sqrt{x}$:
    $g(x)$ is defined only for $x \geq 0$.

Definition of Partial Functions

Definition 2.28

- *Let A, B be sets. A partial function f from A to B, written $f : A \overset{\sim}{\to} B$, is a function $f : A' \to B$ for some $A' \subseteq A$.*
  A' is called the **domain of f**, *written as $A' = \mathrm{dom}(f)$.*
- *Let $f : A \overset{\sim}{\to} B$.*
  - **$f(a)$ is defined**, *written as* **$f(a) \downarrow$**, *if $a \in \mathrm{dom}(f)$.*
  - *Let $b \in \mathbb{N}$.*
    $f(a) \simeq b$ *(***$f(a)$ is partially equal to b***)*

    $:\Leftrightarrow f(a) \downarrow \wedge f(a) = b$.

Terms formed from Partial Functi

- We want to work with terms like $f(g(2), h(3))$, where f, g, h are partial functions.
- **Question:** what happens if $g(2)$ or $h(3)$ is undefined?
  - There is a theory of partial functions, in which $f(g(2), h(3))$ might be defined, even if $g(2)$ or $h(3)$ is undefined.
  - Makes senses for instance for the function $f : \mathbb{N}^2 \overset{\sim}{\to} \mathbb{N}$, $f(x, y) = 0$.
  - Theory of such functions is more complicated.

Strict vs. Non-strict Functions

- Functions, which are defined, even if some of its arguments are undefined, are called **non-strict**.
- Functions, which are defined only if all of its arguments are defined are called **strict**.

Call-By-Value

- **Strict** function are obtained by "**call-by-value**" evaluation.
  - Call-by-value means that before the value of a function applied to arguments, is computed, the arguments of the function are evaluated.
  - If we treat undefinedness as non-termination, then all functions computed by call-by-value will be strict.
    - There is as well finite error, e.g. the error if a division by 0 occurs. This kind of undefinedness will be handled in a non-strict way by many programming languages.
  - Most programming languages (including practially all imperative and object-oriented languages), use call-by-value evaluation.

Call-By-Name

- **Non-strict** functions are obtained by "**call-by-name**" evaluation:
  - The arguments of a function are evaluated only if they are needed in the computation of f.
  - **Haskell** uses **call-by-name-evaluation**.
  - Therefore functions in Haskell are in general **non-strict**.

Example

- Let $f : \mathbb{N}^2 \stackrel{\sim}{\to} \mathbb{N}$, $f(x, y) = x$.
- Let t be an undefined term, e.g. $g(0)$, where $g : \mathbb{N} \stackrel{\sim}{\to} \mathbb{N}$, $g(x) := g(x)$.
  - So the recursion equation of $g(x)$ doesn't terminate.
- With call-by-name, the term $f(2, t)$ evaluates to 2, since we never need to evaluate t.
- With call-by-value, first t is evaluated, which never terminates, so $f(2, t)\uparrow$.
- In our setting, functions are strict, so $f(2, t)$ as above is undefined.

Terms formed from Partial Functi

- In this lecture, functions will always be strict.
- Therefore, a term like $f(g(2), h(3))$ is defined only, if $g(2)$ and $h(3)$ are defined, and if f applied to the results of evaluating $g(2)$ and $h(3)$ is defined.
- $f(g(2), h(3))$ is evaluated as for ordinary functions: We first compute $g(2)$ and $h(3)$, and then evaluate f applied to the results of those computations.

$\perp$

- $\perp$ (pronounced **bottom**) is a term which is always undefined.
- So $\perp\downarrow$ does not hold.

Remark

- Note that variables are always considered as being defined:

$$x\downarrow$$

- One can easily observe

$$t\downarrow \Leftrightarrow \exists x.t \simeq x$$

Terms formed from Partial Functions

Definition 2.29

- *For expressions t formed from constants, $\perp$, variables and partial functions we define whether $t\downarrow$, and whether $t \simeq b$ holds (for a constant b):*
  - *If $t = a$ is a constant, then $t\downarrow$ holds always and $t \simeq b :\Leftrightarrow a = b$.*
  - *If $t = \perp$, then neither $t\downarrow$ not $t \simeq b$ do hold.*
  - *If $t = x$ is a variable, then $t\downarrow$ holds always, $t \simeq b :\Leftrightarrow x = b$.*
  - $$\begin{aligned} f(t_1,\ldots,t_n) \simeq b \quad &:\Leftrightarrow \quad \exists a_1,\ldots,a_n.t_1 \simeq a_1 \wedge \cdots \wedge t_n \simeq a_n \\ &\qquad \wedge f(a_1,\ldots,a_n) \simeq b \quad . \\ f(t_1,\ldots,t_n)\downarrow \quad &:\Leftrightarrow \quad \exists b.f(t_1,\ldots,t_n) \simeq b \end{aligned}$$

Terms formed from Partial Functi

- $s\uparrow :\Leftrightarrow \neg(s\downarrow)$.
- We define for expressions s, t formed from constants and partial functions

$$s \simeq t :\Leftrightarrow (s\downarrow \leftrightarrow t\downarrow) \wedge (s\downarrow \rightarrow \exists a,b.s \simeq a \wedge t \simeq b \wedge a = b)$$

- **t is total** means $t\downarrow$.
- A function $f : A \overset{\sim}{\to} B$ is **total**, iff $\forall a \in A.f(a)\downarrow$ (or, equivalently, $\mathrm{dom}(f) = A$).

Remark:
Total partial functions are ordinary (non-partial) functions.

Quantifiers

Remark:
Quantifiers always range over defined elements.
So by $\exists m.f(n) \simeq m$ we mean: there exists a defined m s.t. $f(n) \simeq m$.

So from $f(n) \simeq g(k)$ we cannot conclude $\exists m.f(n) \simeq m$ unless $g(k)\downarrow$.

Examples

- Assume $f : \mathbb{N} \overset{\sim}{\to} \mathbb{N}$, dom$(f) = \{n \in \mathbb{N} \mid n > 0\}$.
  $f(n) := n - 1$ for $n \in$ dom(f).
- Let $g : \mathbb{N} \overset{\sim}{\to} \mathbb{N}$, dom$(g) = \{0, 1, 2\}$, $g(n) := n + 1$.
  Then:
- $f(1)\downarrow$, $f(0)\uparrow$, $f(1) \simeq 0$, $f(0) \not\simeq n$ for all $n \in \mathbb{N}$.
- $g(\underbrace{f(0)}_{\uparrow})\uparrow$, since $f(0)\uparrow$.
- $g(\underbrace{f(1)}_{\simeq 0})\downarrow$, since $f(1)\downarrow$, $f(1) \simeq 0$, $g(0)\downarrow$.
- $g(\underbrace{f(4)}_{\simeq 3})\uparrow$, since $f(4)\downarrow$, $f(4) \simeq 3$, but $g(3)\uparrow$.

Remark 2.30

Remark 2.30

(a) If a, b are constants, $s \simeq a$, $s \simeq b$, then $a = b$.

(b) For all terms we have $t\downarrow \Leftrightarrow \exists a.t \simeq a$.

(c) $$\begin{aligned} f(t_1, \ldots, t_n)\downarrow \Leftrightarrow \exists a_1, \ldots, a_n.t_1 \simeq a_1 \wedge \cdots \\ \wedge t_n \simeq a_n \\ \wedge f(a_1, \ldots, a_n)\downarrow \quad . \end{aligned}$$

Examples

$f : \mathbb{N} \overset{\sim}{\to} \mathbb{N}$, dom$(f) = \{n \in \mathbb{N} \mid n > 0\}$, $f(n) := n - 1$ for $n \in$ dom(f).
$g : \mathbb{N} \overset{\sim}{\to} \mathbb{N}$, dom$(g) = \{0, 1, 2\}$, $g(n) := n + 1$.

- $\underbrace{g(f(0))}_{\uparrow} \simeq \underbrace{f(0)}_{\uparrow}$, since both expressions are undefined.
- $\underbrace{g(f(1))}_{\simeq 1} \simeq \underbrace{f(g(1))}_{\simeq 1}$, since both sides are defined and equal to 1.
- $\underbrace{g(f(0))}_{\uparrow} \not\simeq \underbrace{f(g(0))}_{\downarrow}$, since the left hand side is undefined, the right hand side is defined.

Examples

$f : \mathbb{N} \overset{\sim}{\to} \mathbb{N}$, dom$(f) = \{n \in \mathbb{N} \mid n > 0\}$, $f(n) := n - 1$ for $n \in$ dom(f).
$g : \mathbb{N} \overset{\sim}{\to} \mathbb{N}$, dom$(g) = \{0, 1, 2\}$, $g(n) := n + 1$.

- $\underbrace{f(f(2))}_{\simeq 0} \not\simeq \underbrace{f(2)}_{\simeq 1}$, since both sides evaluate to different (defined) values.

Examples

$f : \mathbb{N} \overset{\sim}{\to} \mathbb{N}$, dom$(f) = \{n \in \mathbb{N} \mid n > 0\}$, $f(n) := n - 1$ for $n \in$ dom(f).
$g : \mathbb{N} \overset{\sim}{\to} \mathbb{N}$, dom$(g) = \{0, 1, 2\}$, $g(n) := n + 1$.

- $+$, $\cdot$ etc. can be treated as partial functions. So for instance
  - $\underbrace{f(1)}_{\downarrow} + \underbrace{f(2)}_{\downarrow} \downarrow$, since $f(1) \downarrow$, $f(2) \downarrow$, and $+$ is total.
  - $\underbrace{f(1)}_{\simeq 0} + \underbrace{f(2)}_{\simeq 1} \simeq 1$.
  - $\underbrace{f(0)}_{\uparrow} + f(1)\uparrow$, since $f(0)\uparrow$.

Definition

Assume $f : \mathbb{N}^n \overset{\sim}{\to} \mathbb{N}$.

(a) The **range of f**, in short $\mathsf{ran}(f)$ is defined as follows:

$$\mathsf{ran}(f) := \{y \in \mathbb{N} \mid \exists \vec{x}.(f(\vec{x}) \simeq y)\} \enspace .$$

(b) The **graph of f** is the set G_f defined as

$$\mathsf{G}_f := \{(\vec{x}, y) \in \mathbb{N}^{n+1} \mid f(\vec{x}) \simeq y\} \enspace .$$

Remark on G_f

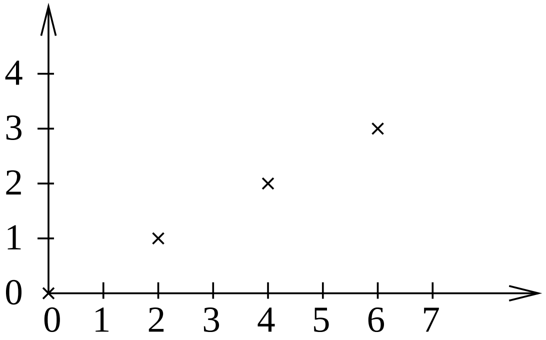
- The notion “graph” used here has nothing to do with the notion of “graph” in graph theory.
- The graph of a function is essentially the graph we draw when visualising f.

Remark on G_f

Example:

$$f : \mathbb{N} \xrightarrow{\sim} \mathbb{N} \ , \qquad f(x) = \begin{cases} \frac{x}{2}, & \text{if } x \text{ even,} \\ \perp, & \text{if } x \text{ is odd.} \end{cases}$$

We can draw f as follows:

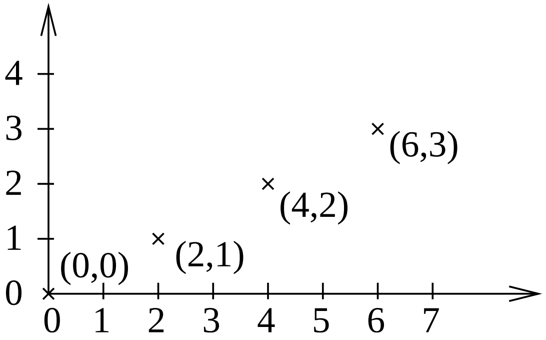


Remark on G_f

In this example we have

$$\mathsf{G}_f = \{(0,0),(2,1),(4,2),(6,3),\ldots\}$$

These are exactly the coordinates of the crosses in the picture:


4
3
2
1
0
(0,0)
(2,1)
(4,2)
(6,3)
0 1 2 3 4 5 6 7