Principles of Abstract Interpretation
MIT press

Ch. **2**, Basic set theory

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com   github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides-02-set-theory-PrAbsInt.pdf

Chapter 2

Ch. **2**, Basic set theory (1/3)

The objective of this Chapter **2** (Basic set theory) is to rapidly review elementary mathematics and fix notations

We have split our review of Chapter **2** into three videos

This first video covers

- numbers
- terms
- predicates
- sets
- relations

If necessary, references are given (in particular to wikipedia) for further deepening your knowledge

Numbers

Numbers

- $\mathbb{N}$: set of all natural numbers (*e.g.*[1] 0, 1, 7, 42)
- $\mathbb{N}^+$: set of all strictly positive natural numbers (*e.g.* 1, 7, 42)
- $\mathbb{Z}$: set of all integer numbers (*e.g.* -42, -7, -1, 0, 1, 7, 42)
- $\mathbb{R}$: set of all real numbers (*e.g.* -3.14, 0, 1, 2.5, π)

en.wikipedia.org/wiki/Natural_number
en.wikipedia.org/wiki/Integer
en.wikipedia.org/wiki/Real_number

[1] *e.g.* stands for Latin *exempli gratia* or example given.

Terms

- Terms are numerical expressions with constants, variables x, y, *etc.*, numerical operators $+$, $-$, $\times$, *etc.*
- Mathematical variables x, x', y, *etc.* denote immutable but unknown entities
- This is *very different* from computer science variables `x`, `y`, *etc.* denoting memories which content is a mutable value
- We write $x \triangleq \mathsf{DEF}$ to mean that the mathematical variable x *denotes* or *is defined as* the term DEF
- For example $2 \triangleq 0 + 1 + 1$

en.wikipedia.org/wiki/Term_(logic)

Predicate logic

Predicates

- $\mathbb{B} \triangleq \{\mathrm{tt}, \mathrm{ff}\}$: set of booleans (tt: true, ff: false)
- Predicates P, Q, *etc*. are statements that are true or false made out of
  - booleans tt, ff
  - boolean variables b, b', $\ldots \in \mathbb{B}$
  - relations ($=$, $\leqslant$, $<$, *etc*.) between terms with variables
  - boolean operators $P \vee Q$ (disjunction), $P \wedge Q$ (conjunction), $\neg P$ (negation), $P \Rightarrow Q$ or $Q \Leftarrow P$ (implication), $P \Leftrightarrow Q$ (if and only if)
  - quantifiers over variables
    - $\exists x \,.\, P(x)$, existential quantifier $\exists$
    - $\forall x \,.\, P(x)$, universal quantifier $\forall$

  (where $P(x)$ makes clear that predicate P depends upon variable x)

en.wikipedia.org/wiki/Predicate_(mathematical_logic)
en.wikipedia.org/wiki/Propositional_calculus
en.wikipedia.org/wiki/First-order_logic

Sets

Sets

- Set S are collections of elements $x \in S$ that *belong to* the set (denoted $\in$)
- $\varnothing$: empty set
- Definition of sets
  - in extension: $S \triangleq \{a, b, c\}$
  - in intention: $S \triangleq \{x \mid p(x)\}$ (or $S' \triangleq \{x \in S \mid q(x)\}$ for subsets)
- We consider a set theory[2] such that
  - sets are built out of an implicitly defined universe $\mathbb{U}$[3]
  - contradictions (like $S \triangleq \{x \mid x \notin S\}$) are forbidden
- $[\ell, u]$: closed interval (similarly $]\ell, u]$, $[\ell, u[$, and $]\ell, u[$)

en.wikipedia.org/wiki/Set_(mathematics)

[2] en.wikipedia.org/wiki/Tarski-Grothendieck_set_theory

[3] en.wikipedia.org/wiki/Grothendieck_universe

Operations on sets

- $\in$: "belongs to"
- $\subseteq$: inclusion, which can be strict $\subsetneq$ ($\supseteq$, $\supsetneq$)
- $|S|$: cardinality of S (number of elements if S is finite)
- $\wp(S)$: powerset (all subsets), $\wp_f(S)$: finite powerset (all finite subsets)
- $S \cup S'$: union or join of sets
- $S \cap S'$: intersection or meet of sets
- $S \setminus S'$: difference of sets
- $\neg S$: complement of a set S with respect to a set U (generally understood from the context) is $\neg S \triangleq U \setminus S$
- $\times$: cartesian product to build tuples $\langle x_1, x_2, \ldots, x_n\rangle$
  en.wikipedia.org/wiki/Set_(mathematics)
  en.wikipedia.org/wiki/Cartesian_product

Relations

Binary relation

- A *binary relation* is $r \in \wp(S \times S')$ a set of pairs $\langle x,\ y\rangle \in r$ of related elements $x \in S$ and $y \in S'$
- Example: $r = \{\langle a,\ b\rangle, \langle a,\ c\rangle\}$ is represented by a graph
$$\begin{tikzcd}[row sep=small] a \arrow[dr, bend left, "r"] & \\ & c \\ b \arrow[ur, bend right, "r"'] & \end{tikzcd}$$
- In $\langle x,\ y\rangle \in r$, x is called the *origin* and y the *extremity*
- $x\ r\ y$ or $x \xrightarrow{r} y$ denotes $\langle x,\ y\rangle \in r$ *e.g.* $1 \leqslant 2$
- $\mathbb{1}_S \triangleq \{\langle x,\ x\rangle \mid x \in S\}$: identity on the set S
- $\mathsf{dom}(r) \triangleq \{x \in S_1 \mid \exists y \in S_2 \mathrel{.} \langle x,\ y\rangle \in r\}$: domain of relation r
- $\mathsf{cod}(r) \triangleq \{y \in S_2 \mid \exists x \in S_1 \mathrel{.} \langle x,\ y\rangle \in r\}$: codomain of r
- $\mathsf{fld}(r) \triangleq \mathsf{dom}(r) \cup \mathsf{cod}(r)$: field of r

`en.wikipedia.org/wiki/Binary_relation`
`simple.wikipedia.org/wiki/Relation_(mathematics)`

Operations on binary relations $r \in \wp(S_1 \times S_2)$

- All operations defined on sets
- $r \rceil S \triangleq \{\langle x,\ y\rangle \in r \mid x \in S\}$: left restriction of r to S
- $r \lceil S \triangleq \{\langle x,\ y\rangle \in r \mid y \in S\}$: right restriction
- $r_1 \fatsemi r_2 \triangleq \{\langle x,\ z\rangle \mid \exists y \,.\, \langle x,\ y\rangle \in r_1 \wedge \langle y,\ z\rangle \in r_2\}$: composition of relations

$$\begin{tikzcd} x \arrow[r, bend left, "r_1"] \arrow[rr, bend right, "r_1 \fatsemi r_2"'] & y \arrow[r, bend left, "r_2"] & z \end{tikzcd}$$

- $r^{-1} \triangleq \{\langle y,\ x\rangle \mid \langle x,\ y\rangle \in r\}$: inverse of relation r

$$\begin{tikzcd} a \arrow[r, bend left, "r"] & b \arrow[l, bend left, "r^{-1}"] \end{tikzcd}$$

en.wikipedia.org/wiki/Composition_of_relations
en.wikipedia.org/wiki/Converse_relation

Mathematical structure of relations

- $\langle \wp(S \times S),\ \fatsemi,\ \mathbb{1}_S \rangle$ is an example of *monoide*
- A *monoide* is a mathematical structure $\langle \mathcal{S},\ \oplus,\ 1 \rangle$ where $\oplus$ is a binary relation on the set $\mathcal{S}$ which is associative (*i.e.* $(x \oplus y) \oplus z = x \oplus (y \oplus z)$) with neutral element 1 (*i.e.* $1 \oplus x = x \oplus 1 = x$).

`en.wikipedia.org/wiki/Monoid`

Properties of relations

- A binary relation $r, \leqslant \in \wp(S \times S)$ is
  - *reflexive* iff $\forall x \in S \,.\, x \; r \; x$
  - *symmetric* iff $\forall x, y \in S \,.\, (x \; r \; y) \Leftrightarrow (y \; r \; x)$
  - *antisymmetric* iff $\forall x, y \in S \,.\, (x \leqslant y \wedge y \leqslant x) \Rightarrow (x = y)$
  - *transitive* iff $\forall x, y, z \in S \,.\, (x \; r \; y \wedge y \; r \; z) \Rightarrow (x \; r \; z)$

- A relation $r \in \wp(S_1 \times S_2)$ is
  - *functional* iff $\forall x \in S_1 \,.\, \forall y, y' \in S_2 \,.\, (\langle x, \; y\rangle \in r \wedge \langle x, \; y'\rangle \in r) \Rightarrow (y = y')$
    In that case we write $r \in \wp_{\scriptscriptstyle F}(S_1 \times S_2)$
  - *total* iff $\forall x \in S_1 \,.\, \exists y \in S_2 \,.\, \langle x, \; y\rangle \in r$

en.wikipedia.org/wiki/Binary_relation
en.wikipedia.org/wiki/Reflexive_relation
en.wikipedia.org/wiki/Symmetric_relation
en.wikipedia.org/wiki/Antisymmetric_relation
en.wikipedia.org/wiki/Transitive_relation

Equivalence relation

- An *equivalence relation* $\equiv$ on a set S is reflexive, symmetric and transitive.
- The equivalence class of a element $x \in S$ is the set $[x]_{\equiv} \triangleq \{y \in S \mid y \equiv x\}$ of all elements of S that are equivalent to x.
- The equivalence classes form a *partition* of S
- The quotient $S|_{\equiv} \triangleq \{[x]_{\equiv} \mid x \in S\}$ is the set of all equivalence classes.

en.wikipedia.org/wiki/Equivalence_relation
en.wikipedia.org/wiki/Equivalence_class
en.wikipedia.org/wiki/Partition_of_a_set

Partial order

- A *partial order* $\leqslant$ on a set S is reflexive, antisymmetric, and transitive.
- The *strict* partial order is $x < y \triangleq (x \leqslant y) \wedge (x \neq y)$.
- An order is *total* if and only if any two elements of S are comparable $(\forall a, b \in S \,.\, (a \leqslant b) \vee (b \leqslant a))$.
- A set S equipped with a partial order $\leqslant$ is called a *poset* $\langle S, \leqslant\rangle$.

`en.wikipedia.org/wiki/Partially_ordered_set`
`en.wikipedia.org/wiki/Total_order`
`en.wikipedia.org/wiki/Partially_ordered_set`

This concludes our reminder on

- numbers
- terms
- predicates
- sets
- relations

from Chapter **2** (Basic set theory)

The End

Principles of Abstract Interpretation
MIT press

Ch. **2**, Basic set theory

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com   github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides-02-set-theory-PrAbsInt.pdf

Chapter 2

Ch. **2**, Basic set theory (2/3)

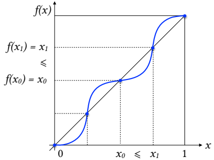
Our next reminder from Chapter **2** (Basic set theory), will be on

- functions
- families
- componentwise definitions

Functions

Partial functions

- A *partial function* $f \in S_1 \nrightarrow S_2$ of S_1 into S_2 is a functional relation r on sets S_1 and S_2 where $f(x)$ denote the unique y such that $\langle x,\ y\rangle \in r$, if it exists.
- If $\forall y \in S_2 \mathrel{.} \langle x,\ y\rangle \notin r$ then f is said to be *undefined* at x
- We sometimes use $f\ x$ or the subscript notation f_x for $f(x)$.
- The set of pairs $\{\langle x,\ f(x)\rangle \mid x \in \mathrm{dom}(f)\}$ is the *function graph*



- We write $f \triangleq x \mapsto\!\!\!\rightarrow e(x)$ when $\forall x \in \mathrm{dom}(f) \mathrel{.} f(x) \triangleq e(x)$
- We write $f \triangleq x \in S \mapsto\!\!\!\rightarrow e(x)$ when $S = \mathrm{dom}(f)$.

en.wikipedia.org/wiki/Partial_function

Total functions

- A *total function* $f \in S_1 \to S_2$ has $\mathrm{dom}(f) = S_1$
- It is everywhere defined on S_1 which we write $x \in S_1 \mapsto f(x)$.
- If $S_1 = S_2 = S$ then $f \in S \to S$ is often called an *operator* on S or an S-*transformer*.
- A function $F \in (S_1 \to S_2) \to (S'_1 \to S'_2)$ taking functions as parameters is called a *functional*.

en.wikipedia.org/wiki/Higher-order_function

Dependent functions

- We write $f \in x \in S_1 \to S_2(x)$ where S_2 maps each $x \in S_1$ to a set $S_2(x)$
- This means that the returned value $f(x)$ of the function f always belong to the $S_2(x)$ set which depends upon one of its parameters x
- Formally, this denotes the set of functions $f \in x \in S_1 \to \bigcup_{x \in S_1} S_2(x)$ such that $\forall x \in S_1 \,.\, f(x) \in S_2(x)$
- Up to an isomorphism $f \in \prod_{x \in S_1} S_2(x)$
- This is called dependent types in computer science
- For example $f \in n \in \mathbb{N} \to \{k \in \mathbb{N} \mid k \geqslant n\}$ specifies a function $f \in \mathbb{N} \to \mathbb{N}$ such that $\forall n \in \mathbb{N} \,.\, f(n) \geqslant n$

en.wikipedia.org/wiki/Dependent_type

Characteristic function

- The characteristic function $\mathbb{c}_S$ of a set S is

$$\mathbb{c}_S \quad \triangleq \quad x \in \mathbb{U} \mapsto (\!| x \in S \mathrel{?} \mathtt{tt} \mathrel{:} \mathtt{ff} |\!)$$

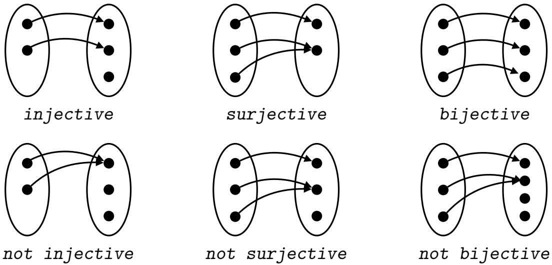
  where $(\!| \ldots \mathrel{?} \ldots \parallel \ldots \mathrel{?} \ldots \parallel \ldots \mathrel{?} \ldots \mathrel{:} \ldots |\!)$ is the conditional expression (as in C)

- The characteristic function of $\{a, b\}$ is $x \in \mathbb{U} \mapsto (\!| x = a \mathrel{?} \mathtt{tt} \parallel x = b \mathrel{?} \mathtt{tt} \mathrel{:} \mathtt{ff} |\!)$

en.wikipedia.org/wiki/Characteristic_function

Properties of functions

- A total function $f \in S_1 \to S_2$ is
  - *injective/one-to-one* iff $\forall x_1 \in S_1 \,.\, \forall x_2 \in S_2 \,.\, x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$ (written $f \in S_1 \rightarrowtail S_2$).
  - *surjective/onto* iff $\forall y \in S_2 \,.\, \exists x \in S_1 \,.\, f(x) = y$ (written $f \in S_1 \twoheadrightarrow S_2$).
  - *bijective* iff both injective and surjective (written $f \in S_1 \rightarrowtail\!\!\!\!\!\twoheadrightarrow S_2$).



en.wikipedia.org/wiki/Surjective_function
en.wikipedia.org/wiki/Injective_function
en.wikipedia.org/wiki/Bijection

Isomorphism

- Sets S_1 and S_2 are *isomorphic* when there exists a bijection of S_1 onto S_2.
- Isomorphic sets have the same *cardinality* (by def. cardinality)
- A set S is *enumerable*, or *denumerable*, or *countable* if and only iff there exists a bijection $\iota \in S \rightarrowtail\!\!\!\!\twoheadrightarrow \mathbb{N}$ between S and the naturals.

en.wikipedia.org/wiki/Isomorphism
en.wikipedia.org/wiki/Cardinality
en.wikipedia.org/wiki/Countable_set

Operations on functions

- The *right image* of a relation $r \in \wp(S_1 \times S_2)$ is the function $x \in S_1 \mapsto \{y \in S_2 \mid \langle x, y\rangle \in r\} \in S_1 \to \wp(S_2)$.
- The *composition* of partial functions is $f \circ g = x \mapsto\!\!\!\rightarrow f(g(x))$.
- Considered are relations, this is $g \mathbin{⨾} f$.

en.wikipedia.org/wiki/Function_composition

Pointwise definitions

- A *pointwise* definition of a relation is

$$\dot{r} \triangleq f, g \mapsto \forall x \,.\, r(f(x), g(x))$$

- For example, $f \mathrel{\dot{\sqsubseteq}} g$ is $\forall x \,.\, f(x) \sqsubseteq g(x)$

- A *functional pointwise definition* is

$$\begin{aligned} \ddot{r} &\triangleq f, g \mapsto \forall X \,.\, \dot{r}(f(X), g(X)) \\ &= f, g \mapsto \forall X \,.\, \forall x \,.\, r(f(X)x, g(X)x) \end{aligned}$$

- For example, $F \mathrel{\ddot{\sqsubseteq}} G$ is $\forall f \,.\, \forall x \,.\, F(f)x \sqsubseteq G(f)x$

- *etc*.

en.wikipedia.org/wiki/Pointwise

Families

Family

- A *family* $F \in \Delta \to S$ of elements of S *indexed by* Δ is a map from a set Δ (called the domain or index set, which may be infinite) into a set S.
- A family defines a set $\{F(i) \mid i \in \Delta\}$ (where $F(i)$ is often denoted F_i with an index $i \in \Delta$).
- A family defines a cartesian product $\prod_{i \in \Delta} F_i$
- A family defines a sequence $\langle F_i,\ i \in \Delta\rangle$ when Δ is totally ordered.

en.wikipedia.org/wiki/Family_of_sets

Componentwise order

- If $\langle\langle L_i, \sqsubseteq_i\rangle, i \in \Delta\rangle$ is a family of posets then the *componentwise order* (or *product order*) $\dot{\sqsubseteq}$ on the cartesian product $\prod_{i\in\Delta} L_i$ is

$$\prod_{i\in\Delta} x_i \mathrel{\dot{\sqsubseteq}} \prod_{i\in\Delta} y_i \quad \triangleq \quad \forall i \in \Delta \mathrel{.} x_i \sqsubseteq_i y_i$$

- The componentwise order $\dot{\sqsubseteq}$ is sometimes denoted $\prod_{i\in\Delta} \sqsubseteq_i$ or $\sqsubseteq_1 \times \sqsubseteq_2$ when $\Delta = \{1, 2\}$.

en.wikipedia.org/wiki/Product_order
en.wikipedia.org/wiki/Pointwise

This concludes our reminder on

- functions
- families
- componentwise definitions

The End

Principles of Abstract Interpretation
MIT press

Ch. **2**, Basic set theory

Patrick Cousot

pcousot.github.io

PrAbsInt@gmail.com github.com/PrAbsInt/

These slides are available at
http://github.com/PrAbsInt/slides/slides-02-set-theory-PrAbsInt.pdf

Chapter 2

Ch. **2**, Basic set theory (3/3)

Our next reminder from Chapter **2** (Basic set theory), will be on

- recursive definitions
- properties
- proofs

Recursive definitions

Recursive definition

- A *recursive object* is defined in terms of itself
- Example of factorial $!0 \triangleq 1$ and $!n \triangleq n \times !(n-1)$
- More generally, $f \in \mathbb{N} \to S$ where S is a set has the form
  - $f(0) \triangleq c$ where $c \in S$
  - $f(n) \triangleq F(n, f(n-1))$ where $F \in \mathbb{N} \times S \to S$

en.wikipedia.org/wiki/Recursion
en.wikipedia.org/wiki/Recursion_(computer_science)
en.wikipedia.org/wiki/Recursive_definition

Well-definedness of definitions

- Recursive definitions may be *ill-defined*
- Example: $f(0) \triangleq 0$ and $f(n) \triangleq f(n+1)$ when $n \neq 0$.

  We have

  - $f(n) = 0$ for $n \leqslant 0$
  - $f(n)$ is undefined when $n > 0$.

- For programs "undefined" means "does not terminate" or "terminates with a runtime error" (such as `Stack overflow` or `Segmentation fault`, *etc*.).

- So recursive definitions must be proved to be *well-defined* (*e.g*. $! \in \mathbb{N} \to \mathbb{N}$)

`en.wikipedia.org/wiki/Well-defined`

Properties

Properties (predicates, assertions, statements, *etc.*) as sets

- We understand properties as the set of mathematical objects that have this property
- "to be an even integer" is $\{x \in \mathbb{Z} \mid \exists k \in \mathbb{Z} \,.\, x = 2k\}$
- Formally, $P \in \wp(\mathbb{U})$ is called a *property*
- if P is a property then
  - $x \in P$ means "x has property P"
  - $x \notin P$ means "x does not have property P"

en.wikipedia.org/wiki/Property
en.wikipedia.org/wiki/Property_(mathematics)
en.wikipedia.org/wiki/Predicate_(mathematical_logic)

Properties are sets

- When considering properties as sets, *logical implication* $\Rightarrow$ is *subset inclusion* $\subseteq$.
- For example "to be greater than 42 implies to be positive" is $\{x \in \mathbb{Z} \mid x > 42\} \subseteq \{x \in \mathbb{Z} \mid x \geqslant 0\}$.
- With characteristic functions:

$$P \subseteq Q \quad \Leftrightarrow \quad \mathbb{c}_P \stackrel{.}{\Rightarrow} \mathbb{c}_Q$$

- If $P \subseteq Q$ then P is said to be *stronger/more precise* than Q and Q is said to be *weaker/less precise* that P.
- Stronger/more precise properties are satisfied by *less* elements while weaker/less precise properties are satisfied by *more* elements.
- $\mathtt{ff}$ *i.e.* $\emptyset$ is the strongest property while $\mathtt{tt}$ *i.e.* $\mathbb{Z}$ is the weakest property of integers.

Proofs

Proofs

- Given an hypothesis P and a conclusion R, a mathematical proof that $P \Rightarrow R$ is a succession of intermediate results $P \Rightarrow Q_0 \Rightarrow Q_1 \Rightarrow \ldots \Rightarrow Q_n \Rightarrow R$ based on arguments considered true in mathematics (axioms, rules of inference, previously proved lemmas, *etc.*)
- Example (Peano arithmetics): proof that $0 + 1 + 1 \in \mathbb{N}$
  - $0 \in \mathbb{N}$ axiom
  - $\dfrac{n \in \mathbb{N}}{n + 1 \in \mathbb{N}}$ rule of inference

$$
\begin{array}{ll}
& \mathtt{tt} \\
\Rightarrow & 0 \in \mathbb{N} \quad \text{(axiom)} \\
\Rightarrow & 0 + 1 \in \mathbb{N} \quad \text{(rule of inference)} \\
\Rightarrow & 0 + 1 + 1 \in \mathbb{N} \quad \text{(rule of inference)} \quad \text{Q.E.D.}
\end{array}
$$

en.wikipedia.org/wiki/Mathematical_proof
en.wikipedia.org/wiki/Peano_axioms

Proof by contraposition

- A proof of $P \Rightarrow Q$ by *contraposition* consists in proving the contrapositive $\neg Q \Rightarrow \neg P$.

  Proof

  If P is true then $\neg P$ is false

  Since $\mathrm{ff} \Rightarrow \mathrm{ff}$ but $\mathrm{tt} \not\Rightarrow \mathrm{ff}$, $\neg Q \Rightarrow \neg P$ and $\neg P$ is false, implies $\neg Q$ is false

  Therefore Q is true. $\square$

`en.wikipedia.org/wiki/Contraposition`
`en.wikipedia.org/wiki/Proof_by_contrapositive`

Proof by reductio ad absurdum or by contradiction

- A proof of P by *reductio ad absurdum* consists in finding a property Q which is known to be true and proving that $\neg P \Rightarrow \neg Q$.
- By contraposition $Q \Rightarrow P$ that is $\mathtt{tt} \Rightarrow P$ and so P is true.

en.wikipedia.org/wiki/Proof_by_contradiction

Proof by recurrence

Theorem 2.13
To prove that a property P holds for all natural numbers *i.e.* $\mathbb{N} \subseteq P$ (equivalently $\forall n \in \mathbb{N} \,.\, n \in P$), the *proof by recurrence* consists in proving

$$0 \in P \qquad \text{basis}$$

$$\forall n \in \mathbb{N} \,.\, (n \in P) \Rightarrow (n + 1 \in P) \qquad \text{induction step}$$

$n \in P$ is called the induction hypothesis or recurrence hypothesis.

So $n + 1 \in P$ must be proved assuming this induction hypothesis.

en.wikipedia.org/wiki/Mathematical_induction

Soundness of the proof by recurrence

- If you made a proof by recurrence then $\mathbb{N} \subseteq P$
  - Assume that we have made the proof by recurrence and $\mathbb{N} \nsubseteq P$.
  - Then $\exists n \in \mathbb{N} \, . \, n \notin P$.
  - The case $n = 0$ is impossible since we proved $0 \in P$.
  - So $n > 0$ hence $n = (n - 1) + 1$.
  - We proved that $\forall m \in \mathbb{N} \, . \, (m \in P) \Rightarrow (m + 1 \in P)$ so $\neg(m + 1 \in P) \Rightarrow \neg(m \in P)$.
  - For $m = n - 1$ we have $n - 1 \notin P$.
  - Going on this way, $n - 2 \notin P$, $n - 3 \notin P$, ..., $0 \notin P$
  - But $0 \notin P$ is in contradiction with the proof that $0 \in P$.
  - By reductio ad absurdum $\neg(\exists n \in \mathbb{N} \, . \, n \notin P)$
    i.e. $\forall n \in \mathbb{N} \, . \, n \in P$.

en.wikipedia.org/wiki/Soundness

Completeness of the proof by recurrence

If $\mathbb{N} \subseteq P$ then this can always be proved by recurrence.

- Assume, by hypothesis, that $\mathbb{N} \subseteq P$
- Let $Q \triangleq P \cap \mathbb{N}$, so $Q \subseteq \mathbb{N}$
- Moreover $\mathbb{N} \subseteq P$, so $\mathbb{N} \subseteq P \cap \mathbb{N} = Q$
- By antisymmetry, $Q = \mathbb{N}$
- So trivially, $0 \in Q$ and $\forall n \in Q \,.\, n + 1 \in Q$.
- Therefore we have $\mathbb{N} \subseteq Q = P \cap \mathbb{N} \subseteq P$.

So $\mathbb{N} \subseteq P$ can be proved by recurrence (maybe with a stronger recurrence hypothesis Q and an additional implication $Q \subseteq P$).

`en.wikipedia.org/wiki/Completeness_(logic)`

Fermat's proof by infinite descent

$$
\begin{array}{lll}
& \forall n \in \mathbb{N} \,.\, \neg P(n) & \\
\Leftrightarrow & \neg P(0) \wedge \forall n \in \mathbb{N} \,.\, (\forall m \in [0, n[\,.\, \neg P(m)) \Rightarrow \neg P(n) & \wr \text{generalized recurrence} \int \\
\Leftrightarrow & \neg P(0) \wedge \forall n \in \mathbb{N}^+ \,.\, (\neg\neg P(n)) \Rightarrow \neg(\forall m \in [0, n[\,.\, \neg P(m)) & \wr \text{contraposition} \int \\
\Leftrightarrow & \neg P(0) \wedge \forall n \in \mathbb{N}^+ \,.\, P(n) \Rightarrow \exists m \in [0, n[\,.\, \neg\neg P(m) & \\
\Leftrightarrow & \neg P(0) \wedge \forall n \in \mathbb{N}^+ \,.\, P(n) \Rightarrow \exists m \in [0, n[\,.\, P(m) &
\end{array}
$$

By contradiction, if $\exists k \,.\, P(k)$ then there is $k_n > k_{n-1} > \ldots > k_1 > k_0$ s.t. $P(k_i)$, $i = n, \ldots, 0$, in contradiction with $\neg P(0)$

en.wikipedia.org/wiki/Proof_by_infinite_descent

This reminder on

- recursive definitions
- properties
- proofs

concludes our review of Chapter **2** (Basic set theory)

Conclusion

Conclusion

- Set theory is the logical basis for all mathematics and computer science.
- Additional topics in set theory will be covered later in the course, when needed.
- For a more formal introduction to set theory, see *e.g*. the *Introduction to Set Theory* [Monk, 1969] of Don Monk or the more recent [Devlin, 1994]

Bibliography

Devlin, Keith (June 24, 1994). *The Joy of Sets: Fundamentals of Contemporary Set Theory*. 2nd ed. Undergraduate Texts in Mathematics. Springer.

Monk, James Donald (1969). *Introduction to Set Theory*. McGraw-Hill. URL: http://euclid.colorado.edu/~monkd/monk11.pdf.

Home work

Read Ch. **2** "Basic set theory" of

Principles of Abstract Interpretation
Patrick Cousot
MIT Press

The End, Thank you