Discrete Mathematics

Sebastian Iwanowski
FH Wedel

Chapter 3: Proof concepts

References:

Iwanowski/Lang 3 (in German)
Epp 3, 4.1-4.4, 7.4
Rosen 1.5, 3.1-3.4, 4.2

3. Proof Concepts

3.1 Glossary of mathematical structures

- **Assumptions: Axioms**

  Axioms are assumptions that need not be proved.
  They are implicite (i.e. mainly not mentioned) prerequesites for a lot of propositions.

- **Notation: Definitions**

  Definitions are simplifying notations. They are neither propositions nor axioms, i.e. neither to be assumed nor to be proven.

- **Propositions: Theorem, Lemma, Corollary**

  Theorems, lemmas and corollaries are true propositions.
  If an issue is a proposition (and not a definition or an axiom) is often easy to see.
  It is much more difficult to prove that it is a true proposition.

- **Proofs**

  Chain of logical implications in order to prove the truth of a proposition. The chain starts with an assertion (in most cases a set of axioms) and ends with the asserted proposition.

3. Proof Concepts

3.1 Glossary of mathematical structures

Peano's set of axioms for the natural numbers:

Given a set ℕ and a successor relation $\sigma \subset \mathbb{N} \times \mathbb{N}$

1) $0 \in \mathbb{N}$
2) The successor relation is a function.
3) The successor relation in injective.
4) 0 is not a successor of a natural number.
5) With a finite number of successive applications of the successor relation to 0 one can generate *each* element of ℕ.

Theorem: Peano's set of axiom is minimal.

The removal of one axiom admits structures satisfying all other axioms, but looking totally different from our notion of ℕ.

3. Proof Concepts

3.2 Mathematical induction

Mathematical induction is a systematic proof concept which is applied very frequently in computer science.

Basic principle (simplest variant):

The issue to prove is a proposition of the form P(n) for an arbitrary $n \in \mathbb{N}$

1) **Base case:** Prove: P(0) holds.
2) **Inductive step:** Prove: P(n) implies P(n+1).

The proof should not show the validity of P(n), but assume it as prerequesite.
To prove is only the validity of P(n+1).

The mathematical induction must hold for all $n \geq 0$ (no restrictions admitted!)

Examples: see assignments

Own practice makes perfect!

3. Proof Concepts

3.2 Mathematical induction

Generalisation of the basic priciple:

The issue to prove is a proposition of the form P(n) for an arbitrary $n \in \mathbb{N}$

1) Base case: Prove: P(0) holds.

2) Inductive step: Prove: One of P(0), ..., P(n) implies P(n+1)

Applications:

1) Prime factorisation (existence):

   Each natural number n > 1 may be factores in a product $p_1 \cdot p_2 \cdot \ldots \cdot p_k$ such that all factors p_i are prime numbers. (Proof by mathematical induction via n)

2) Divisibility proof using the checksum

   Each natural number n is divisible by 3 if and only if its checksum is divisible by 3. (Proof by mathematical induction via n)

3. Proof Concepts

3.2 Mathematical induction

Inductive definitions for functions $\mathbb{N} \rightarrow \mathbb{N}$:

The function is defined in 2 steps:

i. The function is defined for a certain natural number (usually 0 or 1).

ii. A rule is given how to compute the function value of a number from the function value of the predecessor of that number.

Examples:

1) Factorial $n! = 1 \cdot 2 \cdot 3 \cdot \ldots \cdot n$

   i) $0! = 1$
   ii) $n! = n \cdot (n-1)!$

2) Fibonacci numbers F_n

   i) $F_0 = 0 \quad F_1 = 1$
   ii) $F_n = F_{n-1} + F_{n-2}$

3. Proof Concepts

3.2 Mathematical induction

Generalisation: Recursive definitions of arbitrary sets:

The set is defined in 2 steps:

i. Some elements are defined explicitly (*terminal elements*)

ii. Some rules are given how to generate new elements from old elements (*recursion rules*).

Examples:

1) Grammar definitions over finite alphabets

   i) Some words are defined directly (so-called constants made of terminal symbols).
   ii) Production rules define
   how to form new words from existing words of the grammar.

2) Backus-Naur form for the syntax of programming languages
   (will be discussed in other lectures)

3. Proof Concepts

3.2 Mathematical induction

Applications in geometry and graph theory

Example: Map coloring

Definitions:

A **map** is a decomposition of a two-dimensional area into faces (the „countries“) which are confined by one-dimensional curves (the borders).
Some countries may be open to infinity.

An **admissible coloring** of a map is the assignment of colors to each country such that adjacent countries (having a common border, single points are not considered) have different colors.

Theorem:

Each map generated by just n straight lines (resp. n circles) arbitrarily placed in the plane, may be colored by 2 colors.

3. Proof Concepts

3.3 Other proof strategies

Direct proof

$(p \rightarrow q) \land p \Rightarrow q$

modus ponens

Proof by contraposition

$p \rightarrow q \Leftrightarrow \neg q \rightarrow \neg p$

contraposition

Indirect proof (proof by contradiction)

$(\neg p \rightarrow q) \land (\neg p \rightarrow \neg q) \Rightarrow p$

indirect proof

$(\neg p \rightarrow p) \Rightarrow p$

proof by contradiction

$(\neg p \rightarrow \bot) \Rightarrow p$

proof by contradiction

3. Proof Concepts

3.3 Other proof strategies

Equivalence proof

$$p \leftrightarrow q \Leftrightarrow (p \rightarrow q) \wedge (q \rightarrow p)$$

replacing equivalence by implications

Proof by cases

$$((p_1 \vee p_2) \rightarrow p) \wedge (p_1 \vee p_2) \Rightarrow p$$

proof by 2 cases

analogously:
Proof by more than 2 cases

Proof by enumeration (Pidgeonhole principle)

Given **f: M** $\rightarrow$ **N,** where **M,N** are finite.

Then holds: **|M|** > **|N|** $\Rightarrow$ **f** is not injective.