DM549 and DS(K)820
Lecture 15: Structural Induction

Kevin Aguyar Brix
Email: kabrix@imada.sdu.dk

University of Southern Denmark

4+5 November, 2024

Repetition: Existence of Solutions and Inverses

Multiplicative inverses of a modulo m:

- If $\gcd(a, m) = 1$, there exists a unique multiplicative inverse in $\mathbb{Z}_m$.
- If $\gcd(a, m) \neq 1$, there does not exist a multiplicative inverse.

Solutions of the congruence $a \cdot x \equiv b \pmod{m}$:

- If $\gcd(a, m) = 1$, there exists a unique solution in $\mathbb{Z}_m$.
- If $\gcd(a, m) \neq 1$, there may exist a solution.
  - One can derive conditions under which a solution exists, but we will not elaborate further.

Repetition: Congruence Systems

Definition

Let $a_1, \ldots, a_n, m_1, \ldots, m_n \in \mathbb{Z}$ with $m_1, \ldots, m_n \geq 2$, and let $x \in \mathbb{Z}$ be variable. Then

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\ldots \\ x &\equiv a_n \pmod{m_n} \end{aligned}$$

is called a *congruence system* (kongruenssystem)

Remark:

- If $x = n$ is a solution, then $x = n + k \cdot m$ is also a solution for any $k \in \mathbb{Z}$, where m is the least common multiple of $m_1, m_2, \ldots, m_n$.
- The following are equivalent:
  - (i) There exists a solution.
  - (ii) There exists a solution in $\mathbb{Z}_m$.
  - (iii) There exists precisely one solution in $\mathbb{Z}_m$.

Repetition: The Chinese Remainder Theorem

The Chinese Remainder Theorem (Theorem 4.4.2)

Let $a_1, \ldots, a_n \in \mathbb{Z}$, and let $m_1, \ldots, m_n \geq 2$ be integers that are pairwise relatively prime. Then the congruence system

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\ldots \\ x &\equiv a_n \pmod{m_n} \end{aligned}$$

has a unique solution $x \in \mathbb{Z}_m$ where $m = m_1 \cdot m_2 \cdot \cdots \cdot m_n$.

Note on the name: Name is due to Chinese heritage of problems involving systems of linear congruences.

Repetition: Algorithm for Pairwise Coprime Moduli

The previous constructive proof yields the following algorithm for solving congruence systems with pairwise coprime moduli.

Algorithm:

- Let $m = m_1 \cdot m_2 \cdot \dots \cdot m_n$.
- For $k \in \{1, \dots, n\}$:
  - Let $M_k = \frac{m}{m_k}$.
  - Find the multiplicative inverse y_k of M_k modulo m_k (e.g., using the Euclidean Algorithm).
- Return $x = \sum_{k=1}^{n} M_k y_k a_k$.

Remark: If $m_1, \dots, m_n$ are not pairwise prime, a solution to the congruence system *may* exist, but it cannot be computed with the above method! (Why?)

Recall: Recursive Definitions (Lecture 9)

A recursive definition is a self-referential definition, such as:

Definition (Definition 2.4.5)

The Fibonacci Numbers are defined by:

$$
\begin{aligned}
&f_0 = 0, f_1 = 1, && \text{(base step)} \\
&f_n = f_{n-1} + f_{n-2}, \quad \text{for } n \geq 2 && \text{(recursive step).}
\end{aligned}
$$

Today, we will see similar definitions for sets and structures.

Theorem

For all $n \geq 3$, it holds that

$$f_n \geq \varphi^{n-2}.$$

Today, we will see similar induction proofs for the aforementioned definitions.

Bitstrings

Definitions:

- λ is the *empty string* (den tømme streng).
- $\Sigma = \{0, 1\}$ is called the alphabet (alfabetet).

Definition (Definition 5.3.1)

The set of *bitstrings* (bitstrenge) $\Sigma^\star$ is defined as follows:

$$\lambda \in \Sigma^\star, \qquad \text{(base step)}$$

$$B \in \Sigma^\star \wedge b \in \Sigma \Rightarrow Bb \in \Sigma^\star. \qquad \text{(recursive step)}$$

Palindromes

Definition

The set of *palindromes* P is defined as follows:

$$\lambda, 0, 1 \in P,$$
$$B \in P \land b \in \Sigma \Rightarrow bBb \in P.$$

Definition (Alternative Definition)

The set of *palindromes* P is defined as follows:

$$P_1 = \{\lambda, 0, 1\},$$
$$P_i = P_{i-1} \cup \{bBb \mid B \in P_{i-1} \land b \in \Sigma\} \qquad \text{for } i \in \mathbb{Z}^+, i \geq 2,$$
$$P = \bigcup_{i=1}^{\infty} P_i.$$

Another Recursive Definition

Example (Example 5.3.5)

We define S by:

$$3 \in S,$$
$$x, y \in S \Rightarrow x + y \in S.$$

Example (Alternative Definition, Example 5.3.5)

We define S by:

$$\begin{aligned} S_1 &= \{3\}, \\ S_i &= S_{i-1} \cup \{x + y \mid x, y \in S_{i-1}\}, \qquad \text{for } i \in \mathbb{Z}^+, i \geq 2, \\ S &= \bigcup_{i=1}^{\infty} S_i. \end{aligned}$$

Theorem (Example 5.3.10)

For the set S defined above, it holds $S = \{3n \mid n \in \mathbb{Z}^+\}$.

Structural Induction

Suppose you are given a definition for S_i for any $i \in \mathbb{Z}^+$ as before.

Recipe for Proofs by Structural Induction

To show that $P(S_i)$ holds for all $i \geq 1$, prove:

- Basis step: Prove that $P(S_1)$ holds.
- Inductive step: Prove that

$$\underbrace{P(S_i)}_{\text{inductive hypothesis}} \Rightarrow P(S_{i+1})$$

  for all $i \geq 1$.

Remark: One could consider all the variations (starting at m, strong induction, etc.) that we considered for regular induction, but we won't do that here.

Definition of Full Binary Trees

Definition (Definition 5.3.5)

The set of *full binary trees* (fulde binære træer) is defined the following way:

- There is a full binary tree only consisting of a single vertex, its *root*.
- If T_1 and T_2 are disjoint full binary trees, there is a full binary tree, denoted $T_1 \cdot T_2$, consisting of a root vertex r, T_1, T_2, and edges connecting r to both the root of T_1 and root of T_2.

Definition (Definition 5.3.6)

The *height* (højden) $h(T)$ of a full binary tree T is:

- The height of the full binary tree only consisting of a single vertex is 0.
- If $T = T_1 \cdot T_2$ for two full binary trees T_1 and T_2, then $h(T) = 1 + \max(h(T_1), h(T_2))$.

Note: One can define this more formally, but I did not want to add too much formalism here. You will probably learn about it an later lecture.

Theorem about Full Binary Trees

Theorem (Theorem 5.3.2)

For every binary tree T with $n(T)$ vertices, it holds that

$$n(T) \leq 2^{h(T)+1} - 1.$$