

Induktive Beweise und rekursive Definitionen

Vorlesung Logik in der Informatik, HU Berlin

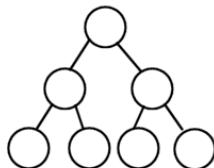
1. Übungsstunde

Beweis durch vollständige Induktion über \mathbb{N}

Aufgabe 1

Zeige, dass für alle $n \in \mathbb{N}$ gilt:

$$\sum_{i=0}^n 2^i = 2^{n+1} - 1.$$

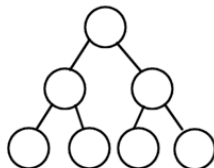


Beweis durch vollständige Induktion über \mathbb{N}

Aufgabe 1

Zeige, dass für alle $n \in \mathbb{N}$ gilt:

$$\sum_{i=0}^n 2^i = 2^{n+1} - 1.$$



Aufgabe 2

Zeige, dass für alle $x \in \mathbb{R}$ mit $x \geq -1$ und alle $n \in \mathbb{N}$ mit $n \geq 1$ gilt:

$$(1+x)^n \geq 1+n \cdot x.$$

Beweis durch vollständige Induktion über \mathbb{N} : Grundidee

Für $n \in \mathbb{N}$ sei $A(n)$ eine Aussage über die Zahl n .

Ziel: Zeige, dass für jedes $n \in \mathbb{N}$ die Aussage $A(n)$ gilt.

Ansatz: Nutze das **Induktionsprinzip**:

Induktionsanfang:

Zeige, dass $A(n)$ für die Zahl $n = 0$ gilt.

Induktionsschritt:

Zeige, dass für jede Zahl $n \in \mathbb{N}$ gilt:

Falls $\underbrace{A(0), \dots, A(n)}_{\text{Induktionsannahme}}$ gelten, so auch $A(n+1)$.

Beweis durch vollständige Induktion über \mathbb{N} : Grundidee

Induktionsanfang:

Zeige, dass $A(n)$ für die Zahl $n = 0$ gilt.

Induktionsschritt:

Zeige, dass für jede Zahl $n \in \mathbb{N}$ gilt:

Falls $A(0), \dots, A(n)$ gelten, so auch $A(n+1)$.

Beweis durch vollständige Induktion über \mathbb{N} : Grundidee

Induktionsanfang:

Zeige, dass $A(n)$ für die Zahl $n = 0$ gilt.

Induktionsschritt:

Zeige, dass für jede Zahl $n \in \mathbb{N}$ gilt:

Falls $A(0), \dots, A(n)$ gelten, so auch $A(n+1)$.

Dann gilt:

$A(0)$ ist wahr gemäß Induktionsanfang.

Beweis durch vollständige Induktion über \mathbb{N} : Grundidee

Induktionsanfang:

Zeige, dass $A(n)$ für die Zahl $n = 0$ gilt.

Induktionsschritt:

Zeige, dass für jede Zahl $n \in \mathbb{N}$ gilt:

Falls $A(0), \dots, A(n)$ gelten, so auch $A(n+1)$.

Dann gilt:

$A(0)$ ist wahr gemäß Induktionsanfang.

$A(1)$ ist wahr da $A(0)$ gilt und wegen dem Induktionsschritt für $n = 0$.

Beweis durch vollständige Induktion über \mathbb{N} : Grundidee

Induktionsanfang:

Zeige, dass $A(n)$ für die Zahl $n = 0$ gilt.

Induktionsschritt:

Zeige, dass für jede Zahl $n \in \mathbb{N}$ gilt:

Falls $A(0), \dots, A(n)$ gelten, so auch $A(n+1)$.

Dann gilt:

$A(0)$ ist wahr gemäß Induktionsanfang.

$A(1)$ ist wahr da $A(0)$ gilt und wegen dem Induktionsschritt für $n = 0$.

$A(2)$ ist wahr da $A(0)$ und $A(1)$ gelten
und wegen dem Induktionsschritt für $n = 1$.

Beweis durch vollständige Induktion über \mathbb{N} : Grundidee

Induktionsanfang:

Zeige, dass $A(n)$ für die Zahl $n = 0$ gilt.

Induktionsschritt:

Zeige, dass für jede Zahl $n \in \mathbb{N}$ gilt:

Falls $A(0), \dots, A(n)$ gelten, so auch $A(n+1)$.

Dann gilt:

$A(0)$ ist wahr gemäß Induktionsanfang.

$A(1)$ ist wahr da $A(0)$ gilt und wegen dem Induktionsschritt für $n = 0$.

$A(2)$ ist wahr da $A(0)$ und $A(1)$ gelten
und wegen dem Induktionsschritt für $n = 1$.

$A(3)$ ist wahr da $A(0)$, $A(1)$ und $A(2)$ gelten
und wegen dem Induktionsschritt für $n = 2$.

Beweis durch vollständige Induktion über \mathbb{N} : Grundidee

Induktionsanfang:

Zeige, dass $A(n)$ für die Zahl $n = 0$ gilt.

Induktionsschritt:

Zeige, dass für jede Zahl $n \in \mathbb{N}$ gilt:

Falls $A(0), \dots, A(n)$ gelten, so auch $A(n+1)$.

Dann gilt:

$A(0)$ ist wahr gemäß Induktionsanfang.

$A(1)$ ist wahr da $A(0)$ gilt und wegen dem Induktionsschritt für $n = 0$.

$A(2)$ ist wahr da $A(0)$ und $A(1)$ gelten
und wegen dem Induktionsschritt für $n = 1$.

$A(3)$ ist wahr da $A(0)$, $A(1)$ und $A(2)$ gelten
und wegen dem Induktionsschritt für $n = 2$.

$A(4)$...

... und so weiter.

Beweis durch vollständige Induktion über \mathbb{N} : Grundidee

Induktionsanfang:

Zeige, dass $A(n)$ für die Zahl $n = 0$ gilt.

Induktionsschritt:

Zeige, dass für jede Zahl $n \in \mathbb{N}$ gilt:

Falls $A(0), \dots, A(n)$ gelten, so auch $A(n+1)$.

Dann gilt:

$A(0)$ ist wahr gemäß Induktionsanfang.

$A(1)$ ist wahr da $A(0)$ gilt und wegen dem Induktionsschritt für $n = 0$.

$A(2)$ ist wahr da $A(0)$ und $A(1)$ gelten
und wegen dem Induktionsschritt für $n = 1$.

$A(3)$ ist wahr da $A(0), A(1)$ und $A(2)$ gelten
und wegen dem Induktionsschritt für $n = 2$.

$A(4) \dots$

\dots und so weiter.

Für alle $n \in \mathbb{N}$ gilt also:

$A(n+1)$ ist wahr da $A(0), \dots, A(n)$ gelten
und wegen dem Induktionsschritt für n .

Rekursive Definition von Mengen

Die Menge L aller Zeichenketten über dem Alphabet

$$A := \{\mathbf{x}, :=, +, -, \neq, ;, \mathbf{while}, \mathbf{do}, \mathbf{end}\} \cup \mathbb{N},$$

die syntaktisch korrekte **WHILE-Programme** sind, ist wie folgt definiert:

Basisregeln:

(B1) Für Zahlen $i, j, c \in \mathbb{N}$ gilt: $\mathbf{x}_i := \mathbf{x}_j + c \in L$.

(B2) Für Zahlen $i, j, c \in \mathbb{N}$ gilt: $\mathbf{x}_i := \mathbf{x}_j - c \in L$.

Rekursive Regeln:

(R1) Sind $w_1 \in L$ und $w_2 \in L$, so ist auch $\mathbf{w}_1; \mathbf{w}_2 \in L$.

(R2) Ist $w \in L$ und $i \in \mathbb{N}$, so ist $\mathbf{while} \mathbf{x}_i \neq 0 \mathbf{do} w \mathbf{end} \in L$.

Rekursive Definition von Mengen

Eine rekursive Definition einer Menge M besteht aus:

Basisregeln der Form " $m \in M$ "

Rekursiven Regeln der Form

“Sind $m_1, \dots, m_k \in M$, dann ist auch $m \in M$ ”,
wobei m von m_1, \dots, m_k abhängt.

Rekursive Definition von Mengen

Zur Erinnerung: Die Menge L aller syntaktisch korrekten WHILE-Programme ist rekursiv wie folgt definiert:

Basisregeln:

(B1) Für Zahlen $i, j, c \in \mathbb{N}$ gilt: $x_i := x_j + c \in L$.

(B2) Für Zahlen $i, j, c \in \mathbb{N}$ gilt: $x_i := x_j - c \in L$.

Rekursive Regeln:

(R1) Sind $w_1 \in L$ und $w_2 \in L$, so ist auch $w_1; w_2 \in L$.

(R2) Ist $w \in L$ und $i \in \mathbb{N}$, so ist **while** $x_i \neq 0$ **do** w **end** $\in L$.

Aufgabe 3:

Welche der folgenden Zeichenketten gehört zur Menge L aller syntaktisch korrekten WHILE-Programme, welche nicht?

(1) $x_3 := x_7 - 2$

(2) $x_3 := 1; x_2 := x_3 + 5$

(3) **while** $x_1 \neq 0$ **do** $x_0 := x_0 + 1; x_1 := x_1 - 1$ **end**

(4) $x_1 := x_1 + 42; \text{while } x_1 \neq 0 \text{ do } x_1 := x_1 - 1$

Rekursive Definition von Funktionen

Sei L wieder die Menge aller syntaktisch korrekten WHILE-Programme.

Aufgabe 4:

Definiere Funktionen $f: L \rightarrow \mathbb{N}$ und $g: L \rightarrow \mathbb{N}$ rekursiv, so dass für alle $w \in L$ gilt:

$$f(w) := \text{Anzahl der " := " in } w,$$

$$g(w) := \text{Anzahl der ";" in } w.$$

Aufgabe 5:

Zeige, dass für alle $w \in L$ gilt:

$$f(w) \geq g(w) + 1.$$

Rekursive Definition von Funktionen

Sei M eine **rekursiv definierte** Menge und sei P eine beliebige Menge.

Die rekursive Definition einer Funktion $f: M \rightarrow P$ sieht folgendermaßen aus:

Rekursionsanfang:

Für jede Basisregel der Form " $m \in M$ " in der Definition von M , definiere $f(m) \in P$.

Rekursionsschritt:

Für jede rekursive Regel der Form

"Sind $m_1, \dots, m_k \in M$, dann ist auch $m \in M$ "

in der Definition von M , definiere $f(m) \in P$ aus $f(m_1), \dots, f(m_k)$.

Rekursive Definition von Funktionen

Notation:

Für eine Menge M schreiben wir 2^M oder $\mathcal{P}(M)$ um die Potenzmenge von M zu bezeichnen, d.h. die Menge aller Teilmengen von M .

Aufgabe 6:

Gib die rekursive Definition einer Funktion $as: AL \rightarrow \mathcal{P}(AS)$ an, so dass für alle $\varphi \in AL$ gilt:

$$as(\varphi) = \{X : X \text{ ist ein Aussagensymbol, das in } \varphi \text{ vorkommt}\}.$$

Rekursive Definition von Funktionen

Notation:

Für eine Menge M schreiben wir 2^M oder $\mathcal{P}(M)$ um die Potenzmenge von M zu bezeichnen, d.h. die Menge aller Teilmengen von M .

Aufgabe 6:

Gib die rekursive Definition einer Funktion $as: AL \rightarrow \mathcal{P}(AL)$ an, so dass für alle $\varphi \in AL$ gilt:

$$as(\varphi) = \{X : X \text{ ist ein Aussagensymbol, das in } \varphi \text{ vorkommt}\}.$$

Aufgabe 7:

(“Koinzidenzlemma für AL ”)

Zeige, dass für alle $\varphi \in AL$ gilt:

Sind $\mathcal{I}_1, \mathcal{I}_2: AS \rightarrow \{0, 1\}$ Interpretationen mit

$$\mathcal{I}_1(X) = \mathcal{I}_2(X) \quad \text{für alle } X \in as(\varphi),$$

dann ist $\llbracket \varphi \rrbracket^{\mathcal{I}_1} = \llbracket \varphi \rrbracket^{\mathcal{I}_2}$.