

Beweistechniken

Vorlesung Logik in der Informatik, HU Berlin

1. Übungsstunde

Was darf in einem Beweis verwendet werden?

- ▶ die Voraussetzungen des Satzes
- ▶ Definitionen und bereits bekannte Tatsachen und Sätze
- ▶ im Beweis selbst oder anderswo bereits als wahr bewiesene Aussagen
- ▶ logische Schlussregeln

Und was ist verboten?

- ▶ unzulässiges Argumentieren mit Beispielen
- ▶ Verwendung gleicher Symbole zur Bezeichnung verschiedener Dinge
- ▶ Hantieren mit nicht exakt oder gar widersprüchlich definierten Begriffsbildungen
- ▶ unzulässige Gedankensprünge beim Schlussfolgern
- ▶ Ausnutzung von bis dahin noch unbewiesenen Behauptungen zur Begründung von einzelnen Beweisschritten

Hilfreiche Beweistechniken

- ▶ direkter Beweis
- ▶ Beweis durch Kontraposition
- ▶ Beweis durch Widerspruch
- ▶ Beweis durch vollständige Induktion
 - ▶ ... über die natürlichen Zahlen
 - ▶ ... über rekursiv definierte Mengen

Direkter Beweis

Ansatz:

Die Behauptung wird “direkt” (d.h. “ohne Umwege”) bewiesen.

Direkter Beweis

Ansatz:

Die Behauptung wird “direkt” (d.h. “ohne Umwege”) bewiesen.

Notation/Erinnerung: Sei $\varphi \in \text{AL}$ und $\mathcal{I} : \text{AS} \rightarrow \{0, 1\}$. Wir schreiben $\mathcal{I} \models \varphi$, falls $\llbracket \varphi \rrbracket^{\mathcal{I}} = 1$ und $\mathcal{I} \not\models \varphi$, falls $\llbracket \varphi \rrbracket^{\mathcal{I}} = 0$.

Direkter Beweis

Ansatz:

Die Behauptung wird “direkt” (d.h. “ohne Umwege”) bewiesen.

Notation/Erinnerung: Sei $\varphi \in \text{AL}$ und $\mathcal{I} : \text{AS} \rightarrow \{0, 1\}$. Wir schreiben $\mathcal{I} \models \varphi$, falls $\llbracket \varphi \rrbracket^{\mathcal{I}} = 1$ und $\mathcal{I} \not\models \varphi$, falls $\llbracket \varphi \rrbracket^{\mathcal{I}} = 0$.

Behauptung 1: Für alle $\varphi, \psi, \chi \in \text{AL}$ und $\mathcal{I} : \text{AS} \rightarrow \{0, 1\}$ gilt:

$$\mathcal{I} \models (\varphi \wedge (\psi \vee \chi)) \iff \mathcal{I} \models ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$$

Direkter Beweis

Ansatz:

Die Behauptung wird “direkt” (d.h. “ohne Umwege”) bewiesen.

Notation/Erinnerung: Sei $\varphi \in \text{AL}$ und $\mathcal{I} : \text{AS} \rightarrow \{0, 1\}$. Wir schreiben $\mathcal{I} \models \varphi$, falls $\llbracket \varphi \rrbracket^{\mathcal{I}} = 1$ und $\mathcal{I} \not\models \varphi$, falls $\llbracket \varphi \rrbracket^{\mathcal{I}} = 0$.

Behauptung 1: Für alle $\varphi, \psi, \chi \in \text{AL}$ und $\mathcal{I} : \text{AS} \rightarrow \{0, 1\}$ gilt:

$$\mathcal{I} \models (\varphi \wedge (\psi \vee \chi)) \iff \mathcal{I} \models ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$$

Behauptung 2: Für alle $\varphi, \psi, \chi \in \text{AL}$ und $\mathcal{I} : \text{AS} \rightarrow \{0, 1\}$ gilt:

$$\mathcal{I} \models (\varphi \vee (\psi \wedge \chi)) \iff \mathcal{I} \models ((\varphi \vee \psi) \wedge (\varphi \vee \chi))$$

Direkter Beweis

Ansatz:

Die Behauptung wird “direkt” (d.h. “ohne Umwege”) bewiesen.

Notation/Erinnerung: Sei $\varphi \in \text{AL}$ und $\mathcal{I} : \text{AS} \rightarrow \{0, 1\}$. Wir schreiben $\mathcal{I} \models \varphi$, falls $\llbracket \varphi \rrbracket^{\mathcal{I}} = 1$ und $\mathcal{I} \not\models \varphi$, falls $\llbracket \varphi \rrbracket^{\mathcal{I}} = 0$.

Behauptung 1: Für alle $\varphi, \psi, \chi \in \text{AL}$ und $\mathcal{I} : \text{AS} \rightarrow \{0, 1\}$ gilt:

$$\mathcal{I} \models (\varphi \wedge (\psi \vee \chi)) \iff \mathcal{I} \models ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$$

Behauptung 2: Für alle $\varphi, \psi, \chi \in \text{AL}$ und $\mathcal{I} : \text{AS} \rightarrow \{0, 1\}$ gilt:

$$\mathcal{I} \models (\varphi \vee (\psi \wedge \chi)) \iff \mathcal{I} \models ((\varphi \vee \psi) \wedge (\varphi \vee \chi))$$

Behauptung 3:

Es gilt für jede Formel $\varphi \in \text{AL}$:

$$\varphi \text{ ist erfüllbar} \iff \neg\varphi \text{ ist nicht allgemeingültig.}$$

Beweis durch Kontraposition

Seien V und A beliebige Aussagen. Dann gilt:

“Falls V gilt, so auch A ” ist wahr.

$\Leftrightarrow$ “ A gilt oder V gilt nicht” ist wahr.

$\Leftrightarrow$ “Falls A nicht gilt, so gilt auch V nicht” ist wahr.

Beweis durch Kontraposition

Seien V und A beliebige Aussagen. Dann gilt:

“Falls V gilt, so auch A ” ist wahr.

$\Leftrightarrow$ “ A gilt oder V gilt nicht” ist wahr.

$\Leftrightarrow$ “Falls A nicht gilt, so gilt auch V nicht” ist wahr.

Ansatz: Beweise einen Satz der Form

“Falls V gilt, so auch A .”

dadurch, zu zeigen dass folgendes gilt:

“Falls A nicht gilt, so kann auch V nicht gelten.”

Beweis durch Kontraposition

Seien V und A beliebige Aussagen. Dann gilt:

“Falls V gilt, so auch A ” ist wahr.

$\Leftrightarrow$ “ A gilt oder V gilt nicht” ist wahr.

$\Leftrightarrow$ “Falls A nicht gilt, so gilt auch V nicht” ist wahr.

Ansatz: Beweise einen Satz der Form

“Falls V gilt, so auch A .”

dadurch, zu zeigen dass folgendes gilt:

“Falls A nicht gilt, so kann auch V nicht gelten.”

Behauptung: Sei $n \in \mathbb{N}_{\geq 1}$ und seien $M_1, \dots, M_n$ endliche Mengen.
Dann gilt für jedes $k \in \mathbb{N}$: Falls

$$|M_1| + \dots + |M_n| > k,$$

so existiert eine Menge $M \in \{M_1, \dots, M_n\}$ mit $|M| > \frac{k}{n}$.

Beweis durch Widerspruch

Ziel:

Beweise einen Satz der Form

Falls die Voraussetzungen **V** erfüllt sind, so gilt auch Aussage **A**.

Beweis durch Widerspruch

Ziel:

Beweise einen Satz der Form

Falls die Voraussetzungen V erfüllt sind, so gilt auch Aussage A .

Ansatz:

1. Nimm an, dass die Voraussetzungen V erfüllt sind, jedoch die Aussage A nicht gilt.
2. Leite daraus einen Widerspruch her.

Beweis durch Widerspruch

Ziel:

Beweise einen Satz der Form

Falls die Voraussetzungen V erfüllt sind, so gilt auch Aussage A .

Ansatz:

1. Nimm an, dass die Voraussetzungen V erfüllt sind, jedoch die Aussage A nicht gilt.
2. Leite daraus einen Widerspruch her.

Behauptung:

Sei ψ eine aussagenlogische Formel.

*Falls $\varphi := (\psi \wedge \neg\psi)$, dann
gibt es keine Interpretation $\mathcal{I} : AS \rightarrow \{0, 1\}$ mit $\mathcal{I} \models \varphi$.*

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.

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 $A(n+1)$.

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 $n = 2$.

$A(4) \dots$ *... und so weiter.*

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

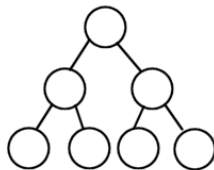
$A(n+1)$ ist wahr da $A(0), \dots, A(n)$ gelten
und wegen dem Induktionsschritt für n .

Beweis durch vollständige Induktion über $\mathbb{N}$

Aufgabe 1

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

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

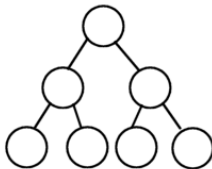


Beweis durch vollständige Induktion über $\mathbb{N}$

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:

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

Rekursive Definition von Mengen

Eine rekursive Definition einer Menge M besteht aus:

Basisregeln der Form “ $m \in M$ ” und

Rekursive Definition von Mengen

Eine rekursive Definition einer Menge M besteht aus:

Basisregeln der Form “ $m \in M$ ” und

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

Die Menge L aller Zeichenketten über dem Alphabet

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

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

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 Definition von Mengen

Die Menge L aller Zeichenketten über dem Alphabet

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

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

Basisregeln:

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

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

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 $xi \neq 0$ do w end** $\in L$.

Rekursive Definition von Mengen

Zur Erinnerung: Definition von L :

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

$\mathbf{while\ x}i \neq 0 \mathbf{\ do\ w\ end} \in L.$

Aufgabe 3:

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

(1) $\mathbf{x4 := x2}$

(2) $\mathbf{x3 := x7 - 2}$

(3) $\mathbf{x3 + 1; x2 := x3 + 5}$

(4) $\mathbf{while\ x1 \neq 0\ do\ x0 := x0 + 1; x1 := x1 - 1\ end}$

(5) $\mathbf{x1 := x1 + 42; while\ x1 \neq 0\ do\ x1 := x1 - 1}$

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:

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 $f(m) \in P$.

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

Zur Erinnerung: Definition von L :

Basisregeln:

(B1) Für Zahlen $i, j, c \in \mathbb{N}$ gilt:

$xi := xj + c \in L$.

(B2) Für Zahlen $i, j, c \in \mathbb{N}$ gilt:

$xi := xj - c \in L$.

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 $xi \neq 0$ do w end $\in L$.

Aufgabe 4:

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

$f(w) :=$ Anzahl der “:=” in w ,

$g(w) :=$ Anzahl der “;” in w .

Rekursive Definition von Funktionen

Zur Erinnerung: Definition von L :

Basisregeln:

(B1) Für Zahlen $i, j, c \in \mathbb{N}$ gilt:

$xi := xj + c \in L$.

(B2) Für Zahlen $i, j, c \in \mathbb{N}$ gilt:

$xi := xj - c \in L$.

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 $xi \neq 0$ do w end $\in L$.

Aufgabe 4:

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

$f(w) :=$ Anzahl der “:=” in w ,

$g(w) :=$ Anzahl der “;” in w .

Aufgabe 5:

Zeige, dass für alle $w \in L$ gilt:

$$f(w) = g(w) + 1.$$

Hinweis: Damit zeigen wir auch, dass (3) in Aufgabe 3 nicht in L ist.

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 $as: AL \rightarrow \mathcal{P}(AS)$ 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 $\llbracket \varphi \rrbracket^{\mathcal{I}_1} = \llbracket \varphi \rrbracket^{\mathcal{I}_2}$.

Kurzeinführung O-Notation

Bei der Analyse von Algorithmen ist man oft an deren Ressourcenverbrauches (Rechenschritte bzw. -zeit oder Speicherplatz) in Abhängigkeit von der Eingabe (bzw. ihrer Größe) interessiert.

Kurzeinführung O-Notation

Bei der Analyse von Algorithmen ist man oft an deren Ressourcenverbrauches (Rechenschritte bzw. -zeit oder Speicherplatz) in Abhängigkeit von der Eingabe (bzw. ihrer Größe) interessiert. Hierfür hat sich die Landau-Notation oder O-Notation durchgesetzt.

Kurzeinführung O-Notation

Bei der Analyse von Algorithmen ist man oft an deren Ressourcenverbrauches (Rechenschritte bzw. -zeit oder Speicherplatz) in Abhängigkeit von der Eingabe (bzw. ihrer Größe) interessiert. Hierfür hat sich die Landau-Notation oder O-Notation durchgesetzt. Dabei beschränkt sich die Analyse auf den asymptotischen Ressourcenverbrauch. D.h.

- ▶ der tatsächliche Ressourcenverbrauch für ein (beliebiges,) endliches Anfangsstück wird ignoriert
- ▶ Konstanten oder asymptotisch kleinere Anstiege der Ressourcen werden ignoriert.

Kurzeinführung O-Notation

Bei der Analyse von Algorithmen ist man oft an deren Ressourcenverbrauches (Rechenschritte bzw. -zeit oder Speicherplatz) in Abhängigkeit von der Eingabe (bzw. ihrer Größe) interessiert. Hierfür hat sich die Landau-Notation oder O-Notation durchgesetzt. Dabei beschränkt sich die Analyse auf den asymptotischen Ressourcenverbrauch. D.h.

- ▶ der tatsächliche Ressourcenverbrauch für ein (beliebiges,) endliches Anfangsstück wird ignoriert
- ▶ Konstanten oder asymptotisch kleinere Anstiege der Ressourcen werden ignoriert.

Beispiel: Sortieren eines n -elementigen Arrays benötigt im worst-case $\approx n \cdot \log(n)$ Rechenschritte. Der Test, ob ein solches Array sortiert ist, benötigt etwa n Schritte. Ein Algorithmus, der erst testet, ob das Array sortiert ist, und, falls nicht, das Array dann sortiert, braucht somit etwa $n \cdot \log(n) + n$ Schritte. Für große n dominiert jedoch der $n \log(n)$ Teil.

Definition O-Notation

Seien f, g Funktionen von $\mathbb{N}$ nach $\mathbb{R}^+ \cup \{0\}$, dann schreiben wir:

Definition O-Notation

Seien f, g Funktionen von $\mathbb{N}$ nach $\mathbb{R}^+ \cup \{0\}$, dann schreiben wir:

$f(n) \in \mathcal{O}(g(n))$, falls es Zahlen $n_0, c \in \mathbb{N}$ gibt, so dass für alle $n \geq n_0$ gilt:

$$f(n) \leq c \cdot g(n).$$

Definition O-Notation

Seien f, g Funktionen von $\mathbb{N}$ nach $\mathbb{R}^+ \cup \{0\}$, dann schreiben wir:

$f(n) \in \mathcal{O}(g(n))$, falls es Zahlen $n_0, c \in \mathbb{N}$ gibt, so dass für alle $n \geq n_0$ gilt:

$$f(n) \leq c \cdot g(n).$$

Formal ist $\mathcal{O}(g(n))$ also die folgende Klasse von Funktionen:

$$\mathcal{O}(g(n)) = \{ f : \mathbb{N} \rightarrow \mathbb{R}^+ \cup \{0\} \mid \text{es ex. } n_0, c \in \mathbb{N} : f(n) \leq c \cdot g(n) \}$$

Definition O-Notation

Seien f, g Funktionen von $\mathbb{N}$ nach $\mathbb{R}^+ \cup \{0\}$, dann schreiben wir:

$f(n) \in \mathcal{O}(g(n))$, falls es Zahlen $n_0, c \in \mathbb{N}$ gibt, so dass für alle $n \geq n_0$ gilt:

$$f(n) \leq c \cdot g(n).$$

Formal ist $\mathcal{O}(g(n))$ also die folgende Klasse von Funktionen:

$$\mathcal{O}(g(n)) = \{f : \mathbb{N} \rightarrow \mathbb{R}^+ \cup \{0\} \mid \text{es ex. } n_0, c \in \mathbb{N} : f(n) \leq c \cdot g(n) \}$$

Die zentrale Aussage hierbei ist : f wächst nicht wesentlich schneller als g .

Definition O-Notation

Seien f, g Funktionen von $\mathbb{N}$ nach $\mathbb{R}^+ \cup \{0\}$, dann schreiben wir:

$f(n) \in \mathcal{O}(g(n))$, falls es Zahlen $n_0, c \in \mathbb{N}$ gibt, so dass für alle $n \geq n_0$ gilt:

$$f(n) \leq c \cdot g(n).$$

Formal ist $\mathcal{O}(g(n))$ also die folgende Klasse von Funktionen:

$$\mathcal{O}(g(n)) = \{f : \mathbb{N} \rightarrow \mathbb{R}^+ \cup \{0\} \mid \text{es ex. } n_0, c \in \mathbb{N} : f(n) \leq c \cdot g(n) \}$$

Die zentrale Aussage hierbei ist : f wächst nicht wesentlich schneller als g .

Zeige: $n \in \mathcal{O}(n^2)$ und $n^k \in \mathcal{O}(2^n)$ für alle $k \in \mathbb{N}$.