



Übung Algorithmen & Datenstrukturen

SS 2016

Kim Völlinger

Übungstermine:

Montag	09:00 - 11:00	RUD 26, 1'303, Marc Bux
Montag	13:00 - 15:00	RUD 26, 1'305, Marc Bux
Montag	13:00 - 15:00	RUD 26, 1'303, Florian Tschorsch
Dienstag	09:00 - 11:00	RUD 26, 1'303, Patrick Schäfer
Dienstag	13:00 - 15:00	RUD 26, 0'313, Kim Völlinger
Mittwoch	09:00 - 11:00	RUD 26, 1'306, Berit Grußien
Donnerstag	13:00 - 15:00	RUD 26, 1'305, Kim Völlinger
Donnerstag	13:00 - 15:00	RUD 26, 0'313, Patrick Schäfer
Freitag	09:00 - 11:00	RUD 26, 1'305, Berit Grußien
Freitag	11:00 - 13:00	RUD 26, 1'305, Florian Tschorsch

Tutoriumstermine:

Montag	17:00 - 19:00	RUD 26, 1'303, Michael R. Jung
Mittwoch	17:00 - 19:00	RUD 26, 1'303, Michael R. Jung
Donnerstag	15:00 - 17:00	RUD 26, 1'306, Michael R. Jung
Freitag	11:00 - 13:00	RUD 25, 3.101, Michael R. Jung

Zulassung zur Klausur

- Übungsschein = 50% aller Punkte (150 Punkte)
- Es gibt 6 Übungsblätter mit je 50 Punkte Punkte
- Abgabe der Aufgaben vor der Vorlesung bis 10:55 oder vorher im Briefkasten im Raum 3.321, RUD 25
- Programmieraufgaben über Goya abgeben
- Programme auf gruenau2 testen
- begründeter Verdacht auf Abschreiben: 0 Punkte
- nicht ausführbares Programm: 0 Punkte

Goya-Gruppen

- 2er Gruppen (in Ausnahmefällen auch 3er Gruppe)
- Mitglieder dürfen aus verschiedenen Übungen sein
- Gruppe muss sich für einen Termin zur Rückgabe der Übungsblätter entscheiden

Goya-Gruppen

- 2er Gruppen (in Ausnahmefällen auch 3er Gruppe)
- Mitglieder dürfen aus verschiedenen Übungen sein
- Gruppe muss sich für einen Termin zur Rückgabe der Übungsblätter entscheiden
- bis zum 1. Mai muss jeder eine Gruppe haben (Goya)
- bis zum 1. Mai muss jeder in einer Übung eingeschrieben sein (Goya)
- ihr könnt dann immer noch einen Wechsel beantragen

Goya-Gruppen

- 2er Gruppen (in Ausnahmefällen auch 3er Gruppe)
- Mitglieder dürfen aus verschiedenen Übungen sein
- Gruppe muss sich für einen Termin zur Rückgabe der Übungsblätter entscheiden
- bis zum 1. Mai muss jeder eine Gruppe haben (Goya)
- bis zum 1. Mai muss jeder in einer Übung eingeschrieben sein (Goya)
- ihr könnt dann immer noch einen Wechsel beantragen
- Gruppennamen möglichst einfach/kurz (den Korrektoren zuliebe)

Goya-Gruppen

- 2er Gruppen (in Ausnahmefällen auch 3er Gruppe)
- Mitglieder dürfen aus verschiedenen Übungen sein
- Gruppe muss sich für einen Termin zur Rückgabe der Übungsblätter entscheiden
- bis zum 1. Mai muss jeder eine Gruppe haben (Goya)
- bis zum 1. Mai muss jeder in einer Übung eingeschrieben sein (Goya)
- ihr könnt dann immer noch einen Wechsel beantragen
- Gruppennamen möglichst einfach/kurz (den Korrektoren zuliebe)
- Wichtige Informationen verschicken wir über Goya

Organisation

- Alle zwei Wochen gibt es ein neues Übungsblatt:
 - 1. Blatt ist online

Organisation

- Alle zwei Wochen gibt es ein neues Übungsblatt:
 - 1. Blatt ist online
- Webseite der Übung (<https://hu.berlin/algodat16>)
 - Übungstermine: falls Übung ausfällt, geht zu einer anderen
 - Übungsblätter
 - Regelungen zur Abgabe
 - Folien & andere Materialien
 - Tutoriumstermine
 - Probeklausur

Inhalt der Übung

Vorbereitung:

KW 17, 18, 20,
22, 24, 26, 28

Wir besprechen ein neues Blatt und
machen vorbereitende Übungen dazu.

Inhalt der Übung

Vorbereitung:

KW 17, 18, 20,
22, 24, 26, 28

Wir besprechen ein neues Blatt und machen vorbereitende Übungen dazu.

Nachbereitung:

KW 19, 21, 23,
25, 27, 29

Wir besprechen die Lösungen eines abgegebenen Blatts.

Vorbereitung heute...

- Besprechung des 1. Übungsblatts
- Landau-Notation
- Pseudocodeanalyse und Algorithmenentwurf

- Eure Vorbereitung zu Hause:
 - Lest das O-Tutorial auf der Webseite der Übung.
 - Wiederholt: Grenzwerte, Ableitungen, Potenz- und Logarithmusgesetze.

Vorbereitung heute...

- Besprechung des 1. Übungsblatts
- Landau-Notation
- Pseudocodeanalyse und Algorithmenentwurf
- Eure Vorbereitung zu Hause:
 - Lest das O-Tutorial auf der Webseite der Übung.
 - Wiederholt: Grenzwerte, Ableitungen, Potenz- und Logarithmusgesetze.

Vorbereitung heute...

- Besprechung des 1. Übungsblatts
- **Landau-Notation**
- Pseudocodeanalyse und Algorithmenentwurf

- Eure Vorbereitung zu Hause:
 - Lest das O-Tutorial auf der Webseite der Übung.
 - Wiederholt: Grenzwerte, Ableitungen, Potenz- und Logarithmusgesetze.

„Groß O“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$O(g) := \{f \mid \exists c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) \leq c * g(n)\}$$

„Groß O“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$O(g) := \{f \mid \exists c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) \leq c * g(n)\}$$

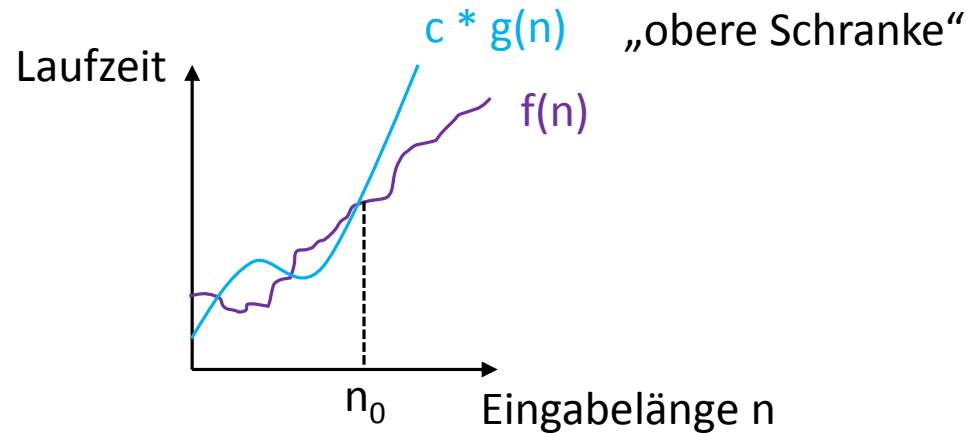
Menge der Fkt., die **nicht wesentlich schneller** wachsen als g .

„Groß O“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$O(g) := \{f \mid \exists c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) \leq c * g(n)\}$$

Menge der Fkt., die **nicht wesentlich schneller** wachsen als g .

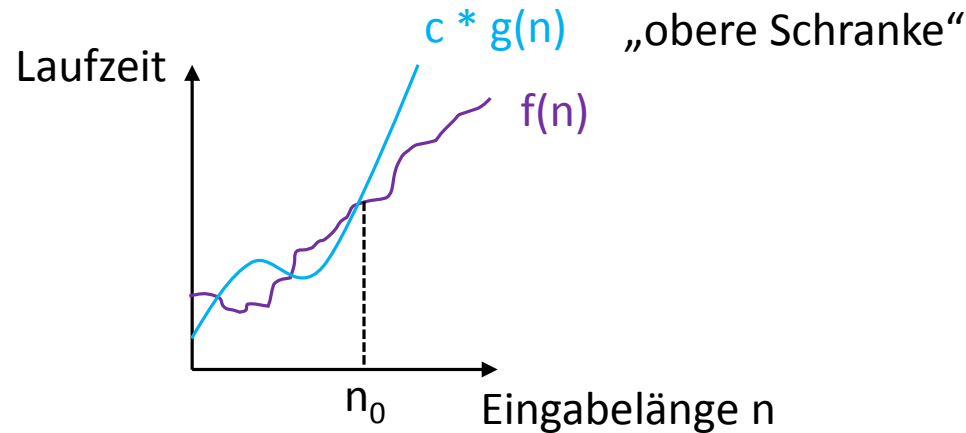


„Groß O“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$O(g) := \{f \mid \exists c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) \leq c * g(n)\}$$

Menge der Fkt., die **nicht wesentlich schneller** wachsen als g .



Hinreichendes Kriterium:

Wenn $\lim_{n \rightarrow \infty} f(n)/g(n) < \infty$,
dann $f \in O(g)$.

Die Funktion im Nenner wächst mind.
so schnell wie die im Zähler.

„Groß Omega“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\Omega(g) := \{f \mid \exists c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) \geq c * g(n)\}$$

„Groß Omega“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\Omega(g) := \{f \mid \exists c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) \geq c * g(n)\}$$

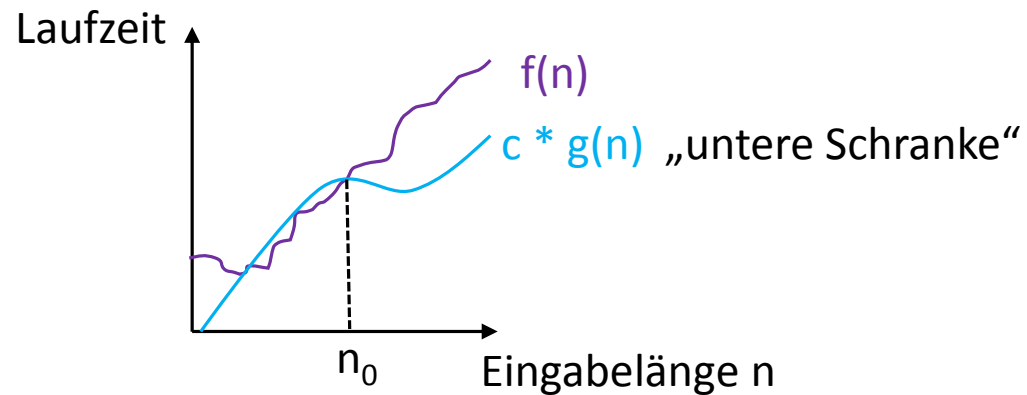
Menge der Fkt., die **nicht wesentlich langsamer** wachsen als g .

„Groß Omega“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\Omega(g) := \{f \mid \exists c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) \geq c * g(n)\}$$

Menge der Fkt., die **nicht wesentlich langsamer** wachsen als g .

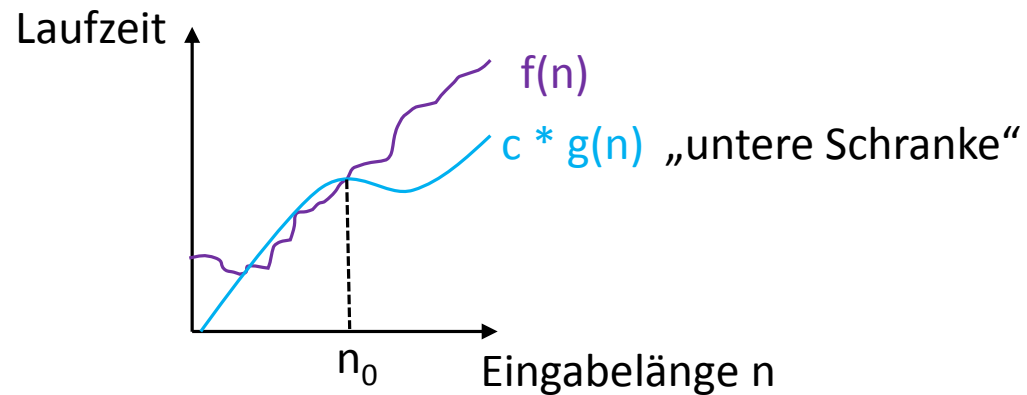


„Groß Omega“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\Omega(g) := \{f \mid \exists c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) \geq c * g(n)\}$$

Menge der Fkt., die **nicht wesentlich langsamer** wachsen als g .



Hinreichendes Kriterium:

Wenn $\lim_{n \rightarrow \infty} g(n)/f(n) < \infty$,
dann $f \in \Omega(g)$.

„(Groß) Theta“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\Theta(g) := \{f \mid \exists c_1, c_2 \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: c_1 * g(n) \leq f(n) \leq c_2 * g(n)\}$$

„(Groß) Theta“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\Theta(g) := \{f \mid \exists c_1, c_2 \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: c_1 * g(n) \leq f(n) \leq c_2 * g(n)\}$$

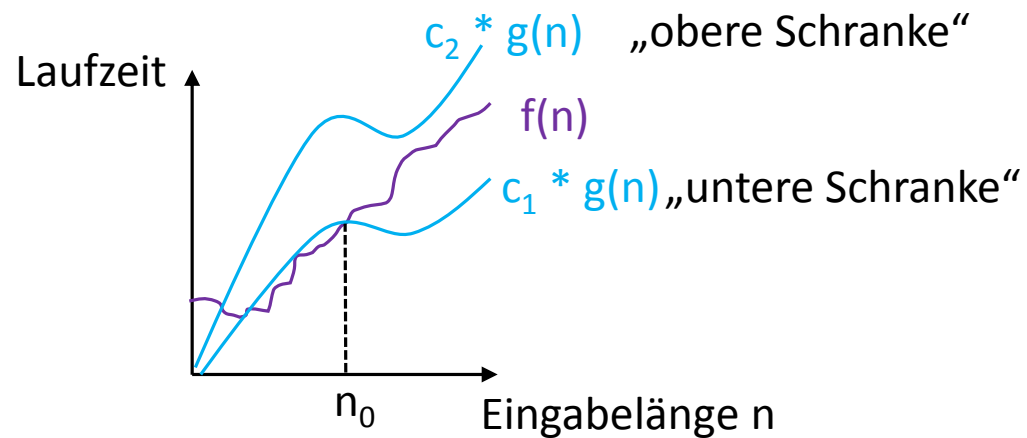
Menge der Fkt., die **genauso schnell** wachsen wie g .

„(Groß) Theta“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\Theta(g) := \{f \mid \exists c_1, c_2 \in \mathbf{R}_{>0} \exists n_0 \in \mathbf{R}_{\geq 0} \forall n \geq n_0: c_1 * g(n) \leq f(n) \leq c_2 * g(n)\}$$

Menge der Fkt., die **genauso schnell** wachsen wie g .

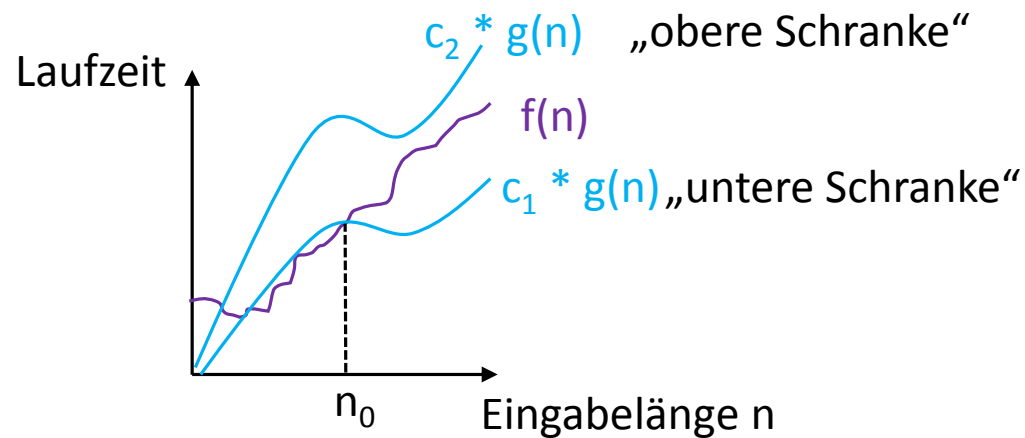


„(Groß) Theta“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\Theta(g) := \{f \mid \exists c_1, c_2 \in \mathbf{R}_{>0} \exists n_0 \in \mathbf{R}_{\geq 0} \forall n \geq n_0: c_1 * g(n) \leq f(n) \leq c_2 * g(n)\}$$

Menge der Fkt., die **genauso schnell** wachsen wie g .



Hinreichendes Kriterium:

Wenn $\lim_{n \rightarrow \infty} f(n)/g(n) < \infty$
und $\lim_{n \rightarrow \infty} g(n)/f(n) < \infty$,
dann $f \in \Theta(g)$.

Denn es gilt: $\Theta(g) = O(g) \cap \Omega(g)$

„klein o“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$o(g) := \{f \mid \forall c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) < c * g(n)\}$$

„klein o“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$o(g) := \{f \mid \forall c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) < c * g(n)\}$$

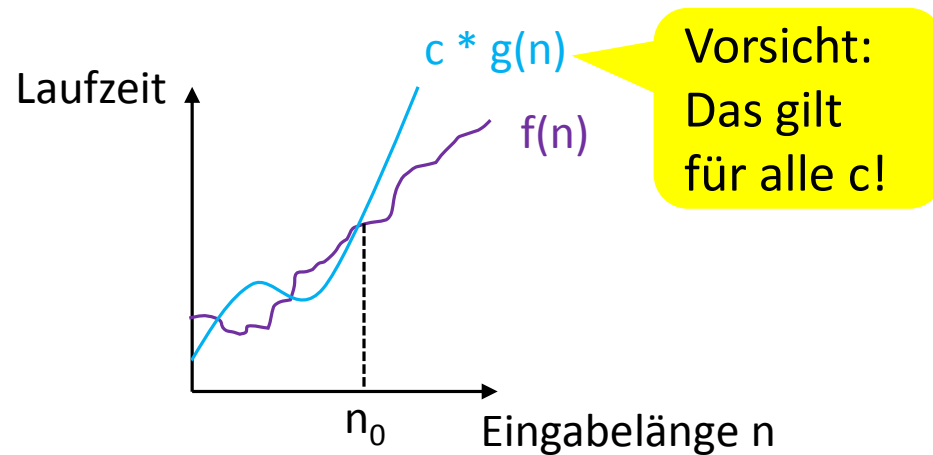
Menge der Fkt., die **langsamer** wachsen als g .

„klein o“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$o(g) := \{f \mid \forall c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) < c * g(n)\}$$

Menge der Fkt., die **langsamer** wachsen als g .

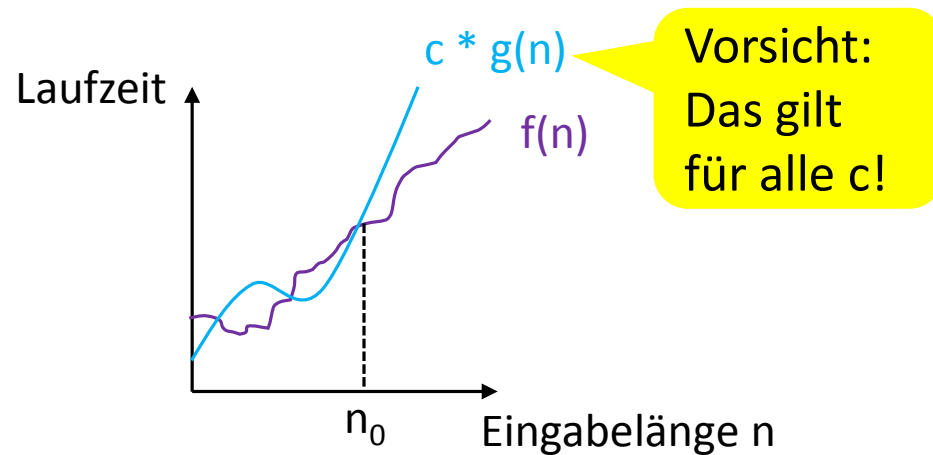


„klein o“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$o(g) := \{f \mid \forall c \in \mathbf{R}_{>0} \ \exists n_0 \in \mathbf{R}_{\geq 0} \ \forall n \geq n_0: f(n) < c * g(n)\}$$

Menge der Fkt., die **langsamer** wachsen als g .



Hinreichendes Kriterium:

Wenn $\lim_{n \rightarrow \infty} f(n)/g(n) = 0$,
dann $f \in o(g)$.

Die Funktion im Nenner wächst
schneller als die im Zähler.

„klein Omega“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\omega(g) := \{f \mid \forall c \in \mathbf{R}_{>0} \exists n_0 \in \mathbf{R}_{\geq 0} \forall n \geq n_0: f(n) > c * g(n)\}$$

„klein Omega“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\omega(g) := \{f \mid \forall c \in \mathbf{R}_{>0} \exists n_0 \in \mathbf{R}_{\geq 0} \forall n \geq n_0: f(n) > c * g(n)\}$$

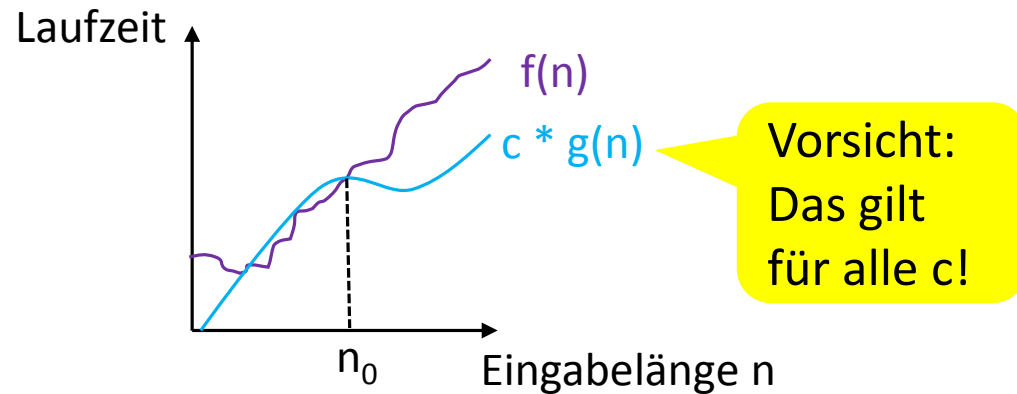
Menge der Fkt., die **schneller** wachsen als g .

„klein Omega“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\omega(g) := \{f \mid \forall c \in \mathbf{R}_{>0} \exists n_0 \in \mathbf{R}_{\geq 0} \forall n \geq n_0: f(n) > c * g(n)\}$$

Menge der Fkt., die **schneller** wachsen als g .

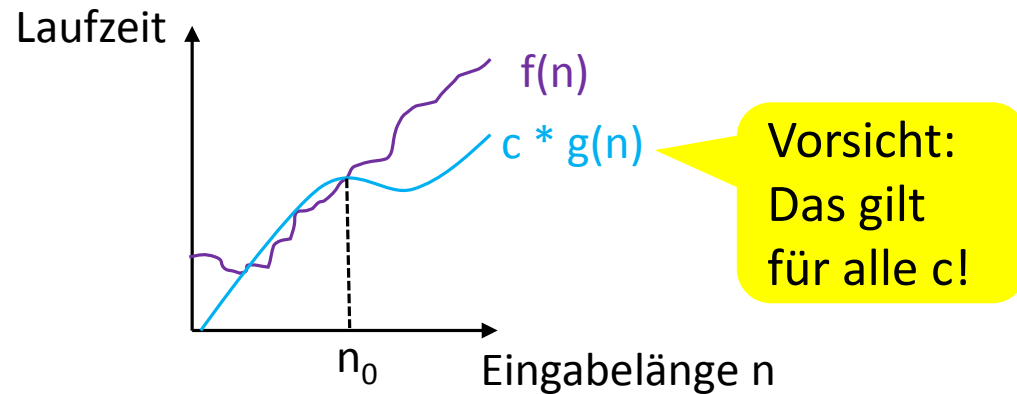


„klein Omega“

Definition: Seien $f, g: \mathbf{R}_{\geq 0} \rightarrow \mathbf{R}_{\geq 0}$

$$\omega(g) := \{f \mid \forall c \in \mathbf{R}_{>0} \exists n_0 \in \mathbf{R}_{\geq 0} \forall n \geq n_0: f(n) > c * g(n)\}$$

Menge der Fkt., die **schneller** wachsen als g .



Hinreichendes Kriterium:

Wenn $\lim_{n \rightarrow \infty} g(n)/f(n) = 0$,
dann $f \in \omega(g)$.

Eigenschaften

„Reflexivität“ von O , Ω , Θ :

$$f \in O(f), f \in \Omega(f), f \in \Theta(f)$$

Eigenschaften

„Reflexivität“ von O , Ω , Θ :

$$f \in O(f), f \in \Omega(f), f \in \Theta(f)$$

„Transitivität“ von O , Ω , Θ , o , ω :

Wenn $f \in O(g)$ und $g \in O(h)$, dann $f \in O(h)$ (analog für Ω , Θ , o , ω)

Eigenschaften

„Reflexivität“ von O , Ω , Θ :

$$f \in O(f), f \in \Omega(f), f \in \Theta(f)$$

„Transitivität“ von O , Ω , Θ , o , ω :

Wenn $f \in O(g)$ und $g \in O(h)$, dann $f \in O(h)$ (analog für Ω , Θ , o , ω)

„Symmetrie“ von Θ :

$$f \in \Theta(g) \text{ gdw. } g \in \Theta(f)$$

Zusammenhänge

- I $f \in O(g)$ gdw. $g \in \Omega(f)$
- II $f \in o(g)$ gdw. $g \in \omega(f)$
- III $f \in \Theta(g)$ gdw. $f \in O(g)$ und $f \in \Omega(g)$

Zusammenhänge

- I $f \in O(g)$ gdw. $g \in \Omega(f)$
- II $f \in o(g)$ gdw. $g \in \omega(f)$
- III $f \in \Theta(g)$ gdw. $f \in O(g)$ und $f \in \Omega(g)$

- IV Wenn $f \in o(g)$, dann $f \notin \Omega(g)$.
- V Wenn $f \in \omega(g)$, dann $f \notin O(g)$.

Zusammenhänge

- I $f \in O(g)$ gdw. $g \in \Omega(f)$
- II $f \in o(g)$ gdw. $g \in \omega(f)$
- III $f \in \Theta(g)$ gdw. $f \in O(g)$ und $f \in \Omega(g)$

- IV Wenn $f \in o(g)$, dann $f \notin \Omega(g)$.
- V Wenn $f \in \omega(g)$, dann $f \notin O(g)$.

- VI $\omega(f) \subseteq \Omega(f)$
- VII $o(f) \subseteq O(f)$
- VIII $\omega(f) \cap o(f) = \{\}$

Vorbereitung heute...

- Besprechung des 1. Übungsblatts
- Landau-Notation
- Pseudocodeanalyse und Algorithmenentwurf
- Eure Vorbereitung zu Hause:
 - Lest das O-Tutorial auf der Webseite der Übung.
 - Wiederholt: Grenzwerte, Ableitungen, Potenz- und Logarithmusgesetze.

Vorbereitung heute...

- Besprechung des 1. Übungsblatts
 - Landau-Notation
 - Pseudocodeanalyse und Algorithmenentwurf
-
- Eure Vorbereitung zu Hause:
 - Lest das O-Tutorial auf der Webseite der Übung.
 - Wiederholt: Grenzwerte, Ableitungen, Potenz- und Logarithmusgesetze.