

Informatik II

Woche 09, 02.03.2017

Giuseppe Accaputo

g@accaputo.ch

Self-Assessment Test

- Dauer: 15 Minuten
- Bitte mein Kürzel **GA** auf Frontseite schreiben
- Falls ihr das Tutorial durchgearbeitet habt, dann schreibt ein **T** auf die Frontseite

Viel Glück!

Self-Assessment Test

- Besprechung der Musterlösung

Repetition: **Java**

Primitive Datentypen

Typ	Grösse	Min. Wert	Max. Wert
byte	8	-128	127
short	16	-32768	32767
int	32	-2^{31}	$2^{31} - 1$
long	64	-2^{63}	$2^{63} - 1$
float	32	$-3.4 \cdot 10^{38}$	$3.4 \cdot 10^{38}$
double	64	$-1.8 \cdot 10^{308}$	$1.8 \cdot 10^{308}$
boolean	n.d.	Entweder true oder false	
char	16	0x0000	0xffff

Arithmetische Zuweisungen

$$a += b \iff a = a + b$$

- Analog für - * /

Konstanten

- Konstanten können mit dem Schlüsselwort `final` definiert werden

```
final double pi = 3.14;  
final double e = 2.718;
```

```
zwei_pi = 2*pi; // Korrekt  
pi = 3.1; // Fehler!
```

Ganzzahldivision

Regeln:

- `int / int = int`
- `int / double = double`
- `double / int = double`

```
int i = 5;  
int j = 3;  
double k = 3.0;  
double res1 = i/j; // res1 = 1.0  
double res2 = i/k; // res2 = 1.666
```

Präzedenz und Assoziativität

Präzedenz	Operatoren	Assoziativität
Unäre Operatoren	! + - ++ --	Links
Arithmetische Operatoren	* / % + -	Links
Schiebe-Operatoren	<< >> >>>	Links
Vergleichs-Operatoren	< <= > >= == !=	Links
Logische Operatoren	& ^ && ?:	Links
Zuweisungsoperatoren	=, op=, wobei op: + - / % & ! ^ << >>	Rechts

Assoziativität

- Linksassoziativ:

$$A \text{ op } B \text{ op } C \iff (A \text{ op } B) \text{ op } C$$

- Rechtsassoziativ:

$$A \text{ op } B \text{ op } C \iff A \text{ op } (B \text{ op } C)$$

Aufgabe Assoziativität

- Setze Klammern um Präzedenz anzuzeigen

```
year % 4 == 0 && year % 100 != 0 || year % 400 == 0
```

Lösung Assoziativität

```
((year % 4) == 0) && ((year % 100) != 0) || ((year % 400) == 0)
```

Faustregel:

1. Punkt vor Strich
2. Arithmetisch vor Vergleich
3. Vergleich vor Logisch
4. ! vor && vor | |

Inkrement und Dekrement Operatoren

- **Prä-Inkrement:**

`y = ++x;` \Leftrightarrow `x = x + 1; y = x;`

- **Post-Inkrement:**

`y = x++;` \Leftrightarrow `y = x; x = x + 1;`

- Analog für Dekrement `--x`

- **Wichtig:** Inkrement und Dekrement Operatoren nur auf Variablen verwenden! Z.B. funktioniert `--4` nicht!

Aufgabe Inkrement Operator

- Was hat **z** für einen Wert nach der Ausführung?

```
int x = 20;  
int y = 30;  
int z = x++ * ++y;
```

Lösung Inkrement Operator

```
int z = x++ * ++y;
```

Präzedenz der
Inkrement Operatoren

```
y = y + 1;  
int z = x * y;  
x = x + 1;
```

```
z == 620
```

Implizite Typkonversion

- Eine *implizite Typkonversion* findet dann statt, wenn der Zieltyp *grösser* ist als der Ursprungstyp:

```
ursprungstyp a = ...;  
zieltyp b = a;
```

- Reihenfolge (aufsteigend in Grösse):

byte < **short** < **int** < **long** < **float** < **double**

Aufgabe Implizite Typkonversion

- Ist dieser Code korrekt?

```
int i = 10213123;  
long l = i ;  
byte b = i ;
```

Lösung Implizite Typkonversion

```
int i = 10213123;  
long l = i ; // OK:      int < long  
byte b = i ; // Fehler:  byte < int
```

Explizite Typkonversion

- Eine *explizite Typkonversion* wird dann benötigt, wenn der Zieltyp *kleiner* ist als der Ursprungstyp:

```
ursprungstyp a = ...;  
zieltyp b = (zieltyp)a;
```

- Reihenfolge (absteigend in Grösse):
double < **float** < **long** < **int** < **short** < **byte**
- Bei expliziter Konversion von **int** nach **float** werden die Nachkommastellen abgeschnitten

Aufgabe Explizite Typkonversion

- Was wird auf der Konsole ausgegeben?

```
System.out.println((double)1/2);
```

Lösung Explizite Typkonversion

```
System.out.println((double)1/2);
```

Konsole:

0.5

Aufgabe Celsius Programm

```
System.out.print("Celsius: ");  
Scanner input = new Scanner(System.in);  
int celsius = input.nextInt();  
  
float fahrenheit = 9 * celsius / 5 + 32;
```

Eingabe:

22.0

Ausgabe:

72.0

Korrekt wäre 72.6!
Wo liegt das Problem?

Lösung Celsius Programm

```
System.out.print("Celsius: ");  
Scanner input = new Scanner(System.in);  
int celsius = input.nextInt();  
  
float fahrenheit = 9f * celsius / 5 + 32;
```

Eingabe:

22.0

Ausgabe:

72.6

Explizite Typkonversion
wird benötigt!

Repetition: *Asymptotisches* Verhalten und Landau Notation

Asymptotische obere Schranke

- Definition:

$$O(g) = \{f: \mathbb{N} \rightarrow \mathbb{R} \mid \exists c > 0, n_0 \in \mathbb{N}: 0 \leq f(n) \leq c \cdot g(n) \quad \forall n \geq n_0\}$$

- Schreibweise:

$$f(n) = n^2 \quad \implies f \in O(n^2)$$

$$h(n) = 2 \cdot n^2 + 3 \quad \implies h \in O(n^2)$$

$$r(n) = n + \sqrt{n} \quad \implies r \in O(?)$$

$$s(n) = n^3 + 300n \quad \implies s \in O(?)$$

Asymptotische obere Schranke

- Definition:

$$O(g) = \{f: \mathbb{N} \rightarrow \mathbb{R} \mid \exists c > 0, n_0 \in \mathbb{N}: 0 \leq f(n) \leq c \cdot g(n) \quad \forall n \geq n_0\}$$

- Schreibweise:

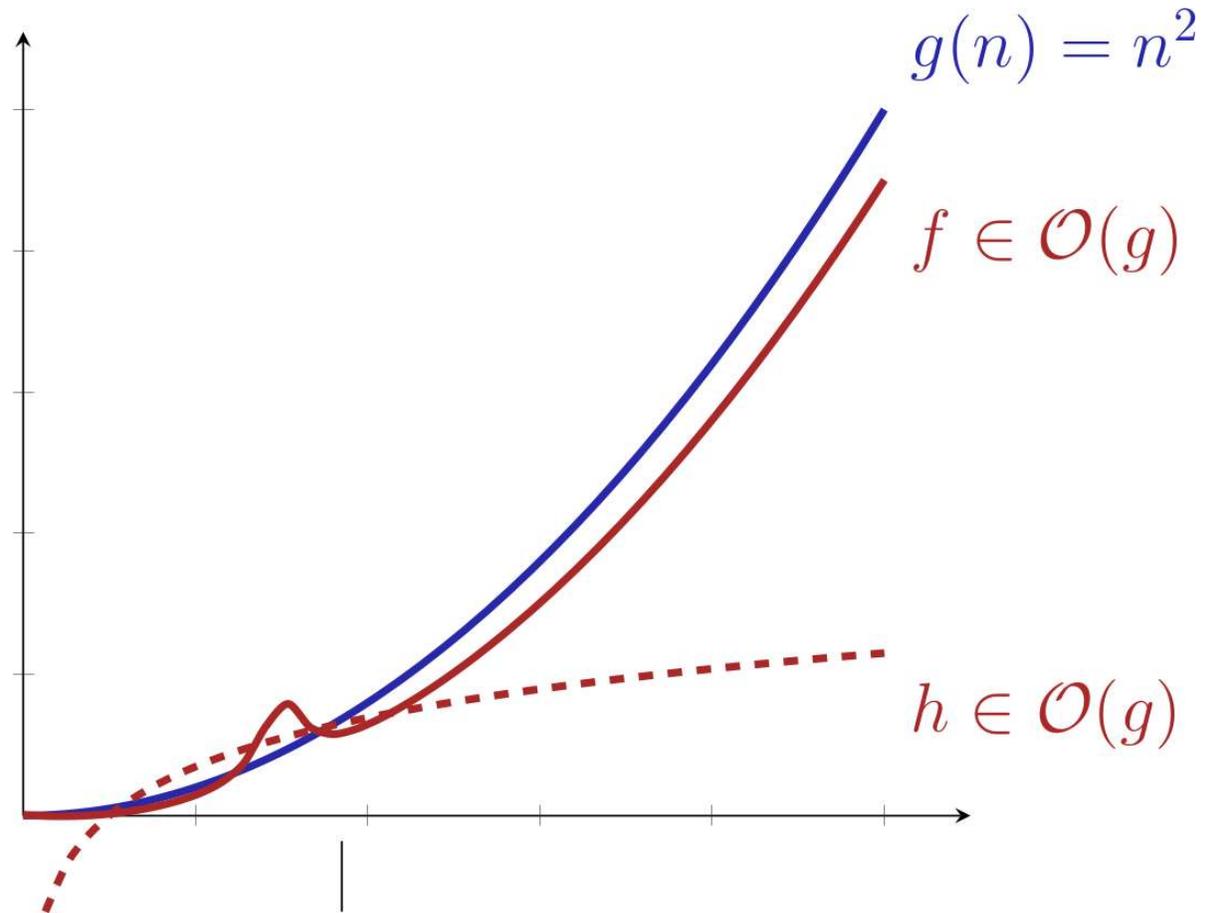
$$f(n) = n^2 \quad \implies f \in O(n^2)$$

$$h(n) = 2 \cdot n^2 + 3 \quad \implies h \in O(n^2)$$

$$r(n) = n + \sqrt{n} \quad \implies r \in O(n)$$

$$s(n) = n^3 + 300n \quad \implies s \in O(n^3)$$

Asymptotische obere Schranke

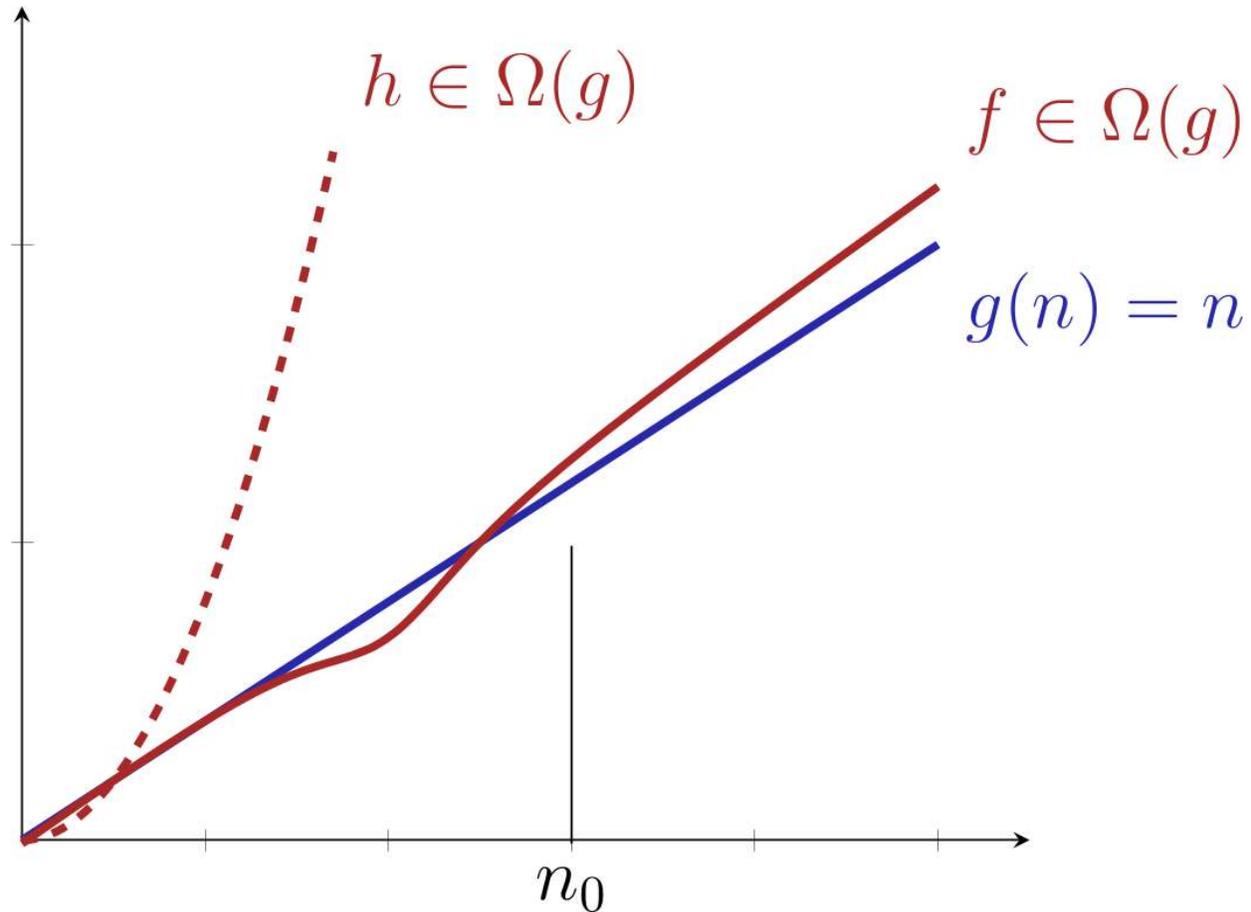


Asymptotische untere Schranke

- Definition:

$$\Omega(g) = \{f: \mathbb{N} \rightarrow \mathbb{R} \mid \exists c > 0, n_0 \in \mathbb{N}: 0 \leq c \cdot g(n) \leq f(n) \quad \forall n \geq n_0\}$$

Asymptotische untere Schranke



Asymptotisch scharfe Schranke

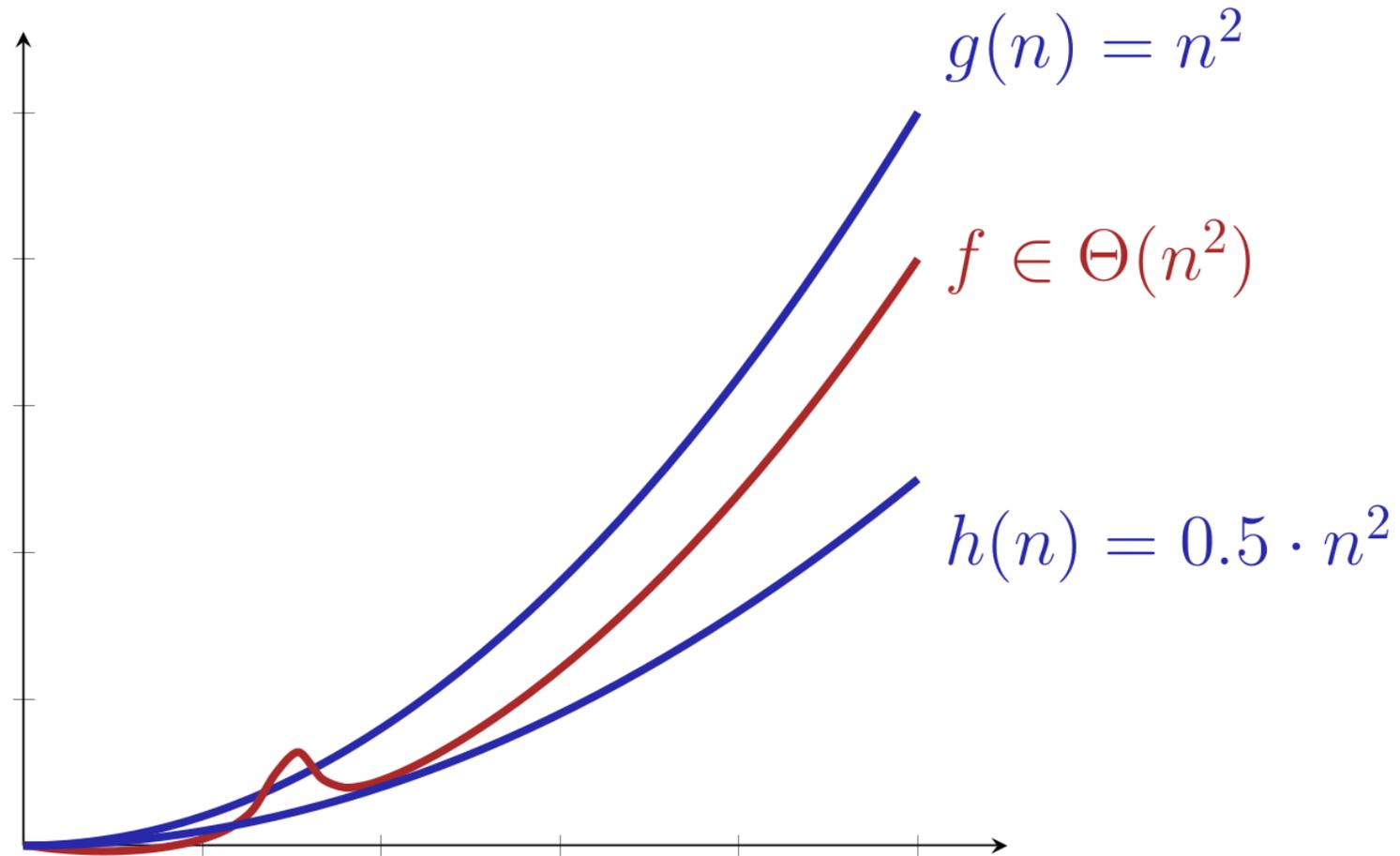
- Definition:

$$\Theta(g) = \Omega(g) \cap O(g)$$

- In der Übung leiten wir die geschlossene Form her

$$\Theta(f) = \{g: \mathbb{N} \rightarrow \mathbb{R} \mid \dots \forall n \geq n_0\}$$

Asymptotisch scharfe Schranke



Aufgabe Asymptotische Komplexität

- Was ist die Komplexität der folgenden Funktion?

```
public static void f1(int a[]){  
    if(a.length != 0)  
        a[0] = 1;  
}
```

Lösung Asymptotische Komplexität

- Was ist die Komplexität der folgenden Funktion?

```
public static void f1(int a[]){  
    if(a.length != 0)  
        a[0] = 1;  
}
```

$O(1)$

Aufgabe Asymptotische Komplexität

- Was ist die Komplexität der folgenden Funktion?

```
public static void f2(int n){  
    int res = 0;  
    for(int i = 0; i<n; i++)  
        res += i;  
}
```

Lösung Asymptotische Komplexität

- Was ist die Komplexität der folgenden Funktion?

```
public static void f2(int n){  
    int res = 0;  
    for(int i = 0; i < n; i++)  
        res += i;  
}
```

$O(n)$

Aufgabe Asymptotische Komplexität

- Was ist die Komplexität der folgenden Funktion?

```
public static void f3(int n){  
    int res = 0;  
    for(int i = 0; i<n; i++)  
        for(int j = 0; j<n; j++)  
            res += j*i;  
}
```

Lösung Asymptotische Komplexität

- Was ist die Komplexität der folgenden Funktion?

```
public static void f3(int n){  
    int res = 0;  
    for(int i = 0; i<n; i++)  
        for(int j = 0; j<n; j++)  
            res += j*i;  
}
```

$$O(n^2)$$

Aufgabe Asymptotische Komplexität

- Was ist die Komplexität der folgenden Funktion?

```
public static void f3(int n){  
    int res = 0;  
    for(int i = 0; i*i<n; i++)  
        res += i;  
}
```

Lösung Asymptotische Komplexität

- Was ist die Komplexität der folgenden Funktion?

```
public static void f3(int n){  
    int res = 0;  
    for(int i = 0; i*i<n; i++)  
        res += i;  
}
```

$$O(\sqrt{n})$$

Vorbesprechung: Übung 2

Fragen oder Anregungen?