Übung "Grundbegriffe der Informatik"

Karlsruher Institut für Technologie

Matthias Schulz, Gebäude 50.34, Raum 247

email: schulz@ira.uka.de

Matthias Janke, Gebäude 50.34, Raum 249

email: matthias.janke@kit.edu

Warshall-Algorithmus

for $k = 0$ **to** $n - 1$ **do**
  for $i = 0$ **to** $n - 1$ **do**
    for $j = 0$ **to** $n - 1$ **do**
     $W[i, j] \leftarrow \max(W[i, j], \min(W[i, k], W[k, j]))$
   od
  od
od

.

Warshall-Algorithmus

for $k = 0$ **to** $n - 1$ **do**
  for $i = 0$ **to** $n - 1$ **do**
    for $j = 0$ **to** $n - 1$ **do**
      $W[i,j] \leftarrow sgn(W[i,j] + W[i,k] \cdot W[k,j])$
    od
  od
od

.

Warshall-Algorithmus

Kürzester Pfad: Eintrag ∞ bei i, j wenn keine Kante.

for $k = 0$ **to** $n - 1$ **do**
  for $i = 0$ **to** $n - 1$ **do**
    for $j = 0$ **to** $n - 1$ **do**
      $W[i, j] \leftarrow min(W[i, j], (W[i, k] + W[k, j]))$
    od
  od
od

.

Warshall-Algorithmus

Kürzester Pfad: Eintrag ∞ bei i, j wenn keine Kante.
$X[i,j] = j$ falls Kante von i nach j, -1 sonst.

for $k = 0$ **to** $n-1$ **do**
  for $i = 0$ **to** $n-1$ **do**
    for $j = 0$ **to** $n-1$ **do**

$$X[i,j] \leftarrow \begin{cases} X[i,j] & \text{falls } W[i,j] < W[i,k] + W[k,j] \\ X[i,k] & \text{sonst.} \end{cases}$$

$$W[i,j] \leftarrow min(W[i,j], (W[i,k] + W[k,j]))$$

    od
  od
od

Warshall-Algorithmus

Feststellung:

Für $k \in \{i, j\}$ gilt

$$\max(W[i,j], \min(W[i,k], W[k,j])) = W[i,j]$$

.

Warshall-Algorithmus

Feststellung:

Für $k \in \{i, j\}$ gilt

$\max(W[i,j], \min(W[i,k], W[k,j])) = W[i,j]$

$k = i$: $\max(W[i,j], \min(W[i,k], W[k,j])) =$
$\max(W[k,j], \min(W[k,k], W[k,j]))$.

$\min(W[k,k], W[k,j]) \leq W[k,j] \Rightarrow \max(W[k,j], \min(W[k,k], W[k,j])) =$
$W[k,j] = W[i,j]$.

.

Warshall-Algorithmus

Feststellung:

Für $k \in \{i, j\}$ gilt

$\max(W[i,j], \min(W[i,k], W[k,j])) = W[i,j]$

$k = j$: $\max(W[i,j], \min(W[i,k], W[k,j])) =$
$\max(W[i,k], \min(W[i,k], W[k,k]))$.

$\min(W[i,k], W[k,k]) \leq W[i,k] \Rightarrow \max(W[i,k], \min(W[i,k], W[k,k])) =$
$W[i,k] = W[i,j]$.

.

Warshall-Algorithmus

Folgerung: k-te Zeile und k-te Spalte ändern sich nicht.

.

Warshall-Algorithmus

Folgerung: k-te Zeile und k-te Spalte ändern sich nicht.

Folgerung: Für die Berechnung kann man alte Werte für $W[i,k], W[k,j]$ verwenden.

.

Warshall-Algorithmus

Berechnen einer Zeile:

$k = 1, i = 2$

$$\begin{pmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{pmatrix} \rightarrow \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{pmatrix}$$

.

Warshall-Algorithmus

Berechnen einer Zeile:

$k = 1, i = 2$

$$\begin{pmatrix} x & x & x & x \\ {\color{red}x} & x & x & x \\ x & {\color{red}x} & x & x \\ x & x & x & x \end{pmatrix} \rightarrow \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ {\color{red}x} & x & x & x \\ x & x & x & x \end{pmatrix}$$

.

Warshall-Algorithmus

Berechnen einer Zeile:

$k = 1, i = 2$

$$\begin{pmatrix} x & x & x & x \\ x & \color{red}{x} & x & x \\ x & \color{red}{x} & x & x \\ x & x & x & x \end{pmatrix} \rightarrow \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ x & \color{red}{x} & x & x \\ x & x & x & x \end{pmatrix}$$

.

Warshall-Algorithmus

Berechnen einer Zeile:

$k = 1, i = 2$

$$\begin{pmatrix} x & x & x & x \\ x & x & \color{red}{x} & x \\ x & \color{red}{x} & x & x \\ x & x & x & x \end{pmatrix} \rightarrow \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ x & x & \color{red}{x} & x \\ x & x & x & x \end{pmatrix}$$

.

Warshall-Algorithmus

Berechnen einer Zeile:

$k = 1, i = 2$

$$\begin{pmatrix} x & x & x & x \\ x & x & x & \color{red}{x} \\ x & \color{red}{x} & x & x \\ x & x & x & x \end{pmatrix} \rightarrow \begin{pmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & \color{red}{x} \\ x & x & x & x \end{pmatrix}$$

.

O-Kalkül

Scheinbar gern benutzter Merksatz:

$g(n) \in O(f(n))$ bedeutet, dass $f(n)$ schneller wächst als $g(n)$.

.

O-Kalkül

Scheinbar gern benutzter Merksatz:

$g(n) \in O(f(n))$ bedeutet, dass $f(n)$ schneller wächst als $g(n)$.

Dies führt zu Überlegungen wie:

n^2 wächst langsamer als $100n^2$, also gilt $100n^2 \notin O(n^2)$

.

O-Kalkül

Scheinbar gern benutzter Merksatz:

$g(n) \in O(f(n))$ bedeutet, dass $f(n)$ schneller wächst als $g(n)$.

Dies führt zu Überlegungen wie:

n^2 wächst langsamer als $100n^2$, also gilt $100n^2 \notin O(n^2)$

Das. Ist. Falsch.

.

O-Kalkül

Besser: $f(n) \in O(g(n))$ bedeutet, dass es eine Konstante c gibt, so dass $f(n)$ höchstens so schnell wächst wie $cg(n)$.

.

O-Kalkül

Besser: $f(n) \in O(g(n))$ bedeutet, dass es eine Konstante c gibt, so dass $f(n)$ höchstens so schnell wächst wie $cg(n)$.

- Was muss für c gelten?

.

O-Kalkül

Besser: $f(n) \in O(g(n))$ bedeutet, dass es eine Konstante c gibt, so dass $f(n)$ höchstens so schnell wächst wie $cg(n)$.

- Was muss für c gelten?

- Für welche n soll die Aussage gelten?

.

O-Kalkül

Besser: $f(n) \in O(g(n))$ bedeutet, dass es eine Konstante c gibt, so dass $f(n)$ höchstens so schnell wächst wie $cg(n)$.

- Was muss für c gelten? $c > 0$

- Für welche n soll die Aussage gelten? $\forall n \geq n_0$

.

O-Kalkül

Besser: $f(n) \in O(g(n))$ bedeutet, dass es eine Konstante c gibt, so dass $f(n)$ höchstens so schnell wächst wie $cg(n)$.

- Was muss für c gelten? $c > 0$

- Für welche n soll die Aussage gelten? Für “hinreichend große” n.

.

O-Kalkül

$$O(f_1(n)) \cdot O(f_2(n)) = O((f_1 \cdot f_2)(n))$$

.

O-Kalkül

$$O(f_1(n)) \cdot O(f_2(n)) = O((f_1 \cdot f_2)(n))$$

Sei $g(n) \in O(f_1(n)) \cdot O(f_2(n))$
$\Rightarrow \exists g_1(n) \in O(f_1(n)) : \exists g_2(n) \in O(f_2(n)) :$
$g(n) = g_1(n) \cdot g_2(n)$

.

O-Kalkül

$O(f_1(n)) \cdot O(f_2(n)) = O((f_1 \cdot f_2)(n))$

Sei $g(n) \in O(f_1(n)) \cdot O(f_2(n))$
$\Rightarrow \exists g_1(n) \in O(f_1(n)) : \exists g_2(n) \in O(f_2(n)) :$
$g(n) = g_1(n) \cdot g_2(n)$

$\exists c_1, c_2 > 0 : \exists n_{01}, n_{02} \in \mathbb{N}_0 :$
$\forall i \in \{1, 2\} : \forall n \geq n_{0i} : g_i(n) \leq c_i f_i(n)$

.

$O(f_1(n)) \cdot O(f_2(n)) = O((f_1 \cdot f_2)(n))$

Sei $g(n) \in O(f_1(n)) \cdot O(f_2(n))$
$\Rightarrow \exists g_1(n) \in O(f_1(n)) : \exists g_2(n) \in O(f_2(n)) :$
$g(n) = g_1(n) \cdot g_2(n)$

$\exists c_1, c_2 > 0 : \exists n_{01}, n_{02} \in \mathbb{N}_0 :$
$\forall i \in \{1, 2\} : \forall n \geq n_{0i} : g_i(n) \leq c_i f_i(n)$

$\Rightarrow \forall n \geq \max(n_{01}, n_{02}) : g_1(n) \cdot g_2(n) \leq c_1 f_1(n) \cdot c_2 f_2(n)$
$= (c_1 c_2) f_1(n) \cdot f_2(n)$, **da alle vorkommenden Zahlen größer oder gleich 0 sind!**

.

O-Kalkül

$$O(f_1(n)) \cdot O(f_2(n)) = O((f_1 \cdot f_2)(n))$$

Sei $g(n) \in O(f_1(n) \cdot f_2(n))$
$\Rightarrow \exists c > 0 : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \leq c f_1(n) \cdot f_2(n)$

O-Kalkül

$$O(f_1(n)) \cdot O(f_2(n)) = O((f_1 \cdot f_2)(n))$$

Sei $g(n) \in O(f_1(n) \cdot f_2(n))$
$\Rightarrow \exists c > 0 : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \leq c f_1(n) \cdot f_2(n)$

Wir setzen

$$g_1(n) = \begin{cases} 1 & \text{falls } n < n_0 \\ c f_1(n) & \text{sonst.} \end{cases}$$

O-Kalkül

$$O(f_1(n)) \cdot O(f_2(n)) = O((f_1 \cdot f_2)(n))$$

Sei $g(n) \in O(f_1(n) \cdot f_2(n))$
$\Rightarrow \exists c > 0 : \exists n_0 \in \mathbb{N}_0 : \forall n \geq n_0 : g(n) \leq c f_1(n) \cdot f_2(n)$

Wir setzen

$$g_1(n) = \begin{cases} 1 & \text{falls } n < n_0 \\ c f_1(n) & \text{sonst.} \end{cases}$$

und

$$g_2(n) = \begin{cases} g(n)/g_1(n) & \text{falls } g_1(n) \neq 0 \\ 0 & \text{sonst.} \end{cases}$$

O-Kalkül

Es gilt: $\forall n \geq n_0 : (g_1(n) = 0 \iff cf_1(n) = 0) \wedge (cf_1(n) = 0 \Rightarrow g(n) \leq 0 \Rightarrow g(n) = 0)$

.

O-Kalkül

Es gilt: $\forall n \geq n_0 : (g_1(n) = 0 \iff cf_1(n) = 0) \wedge (cf_1(n) = 0 \Rightarrow g(n) \leq 0 \Rightarrow g(n) = 0)$

Also gilt für alle $n \in \mathbb{N}_0 : g_1(n) \cdot g_2(n) = g(n)$

.

O-Kalkül

Es gilt: $\forall n \geq n_0 : (g_1(n) = 0 \iff cf_1(n) = 0) \wedge (cf_1(n) = 0 \Rightarrow g(n) \leq 0 \Rightarrow g(n) = 0)$

Also gilt für alle $n \in \mathbb{N}_0 : g_1(n) \cdot g_2(n) = g(n)$

Außerdem gilt: $\forall n \geq n_0 : g_1(n) \leq cf_1(n) \Rightarrow g_1 \in O(f_1)$ und $\forall n \geq n_0 : g_2(n) \leq f_2(n)$.

.

O-Kalkül

Es gilt: $\forall n \geq n_0 : (g_1(n) = 0 \iff cf_1(n) = 0) \wedge (cf_1(n) = 0 \Rightarrow g(n) \leq 0 \Rightarrow g(n) = 0)$

Also gilt für alle $n \in \mathbb{N}_0 : g_1(n) \cdot g_2(n) = g(n)$

Außerdem gilt: $\forall n \geq n_0 : g_1(n) \leq cf_1(n) \Rightarrow g_1 \in O(f_1)$ und $\forall n \geq n_0 : g_2(n) \leq f_2(n)$.

$g_2(n) = 0 \Rightarrow g_2(n) \leq f_2(n) \checkmark$

.

O-Kalkül

Es gilt: $\forall n \geq n_0 : (g_1(n) = 0 \iff cf_1(n) = 0) \wedge (cf_1(n) = 0 \Rightarrow g(n) \leq 0 \Rightarrow g(n) = 0)$

Also gilt für alle $n \in \mathbb{N}_0 : g_1(n) \cdot g_2(n) = g(n)$

Außerdem gilt: $\forall n \geq n_0 : g_1(n) \leq cf_1(n) \Rightarrow g_1 \in O(f_1)$ und $\forall n \geq n_0 : g_2(n) \leq f_2(n)$.

$g_2(n) = 0 \Rightarrow g_2(n) \leq f_2(n)\checkmark$
$g_2(n) \neq 0 \Rightarrow g_2(n) = g(n)/g_1(n)$
$\leq cf_1(n)f_2(n)/(cf_1(n)) = f_2(n)\checkmark$.

.

$2^n \in \Theta(2,1^n)$

.

O-Kalkül

$2^n \in \Theta(2,1^n)$

$2,1^n / 2^n = (1,05)^n$

.

O-Kalkül

$2^n \in \Theta(2,1^n)$

$2,1^n / 2^n = (1,05)^n$

Negation (vom Ω-Teil): $\forall c > 0 \forall n_0 \in \mathbb{N}_0 : \exists n \geq n_0 :$
$2^n \leq c \cdot 2,1^n$

.

O-Kalkül

$2^n \in \Theta(2,1^n)$

$2,1^n/2^n = (1,05)^n$

Negation (vom Ω-Teil): $\forall c > 0 \forall n_0 \in \mathbb{N}_0 : \exists n \geq n_0 :$
$2^n \leq c \cdot 2,1^n$

Sei $c > 0$ beliebig, $n_0 \in \mathbb{N}_0$ beliebig,
$n \geq n_0$ mit $(1,05)^{-n} < c$.

Dann gilt: $c \cdot 2,1^n \geq (1,05)^{-n} \cdot 2,1^n = (2,1/1,05)^n = 2^n$.

.

Allgemein: Es gibt eine Zahlenfolge $(n_1, n_2, \ldots)$, so dass $f(n_i)/g(n_i)$ unbegrenzt und monoton wächst
$\Rightarrow f(n) \notin O(g(n))$

.

Allgemein: Es gibt eine Zahlenfolge $(n_1, n_2, \ldots)$, so dass $f(n_i)/g(n_i)$ unbegrenzt und monoton wächst
$\Rightarrow f(n) \notin O(g(n))$

Wähle zu festem $c > 0$ ein $i \in \mathbb{N}_0$ so, dass $f(n_i)/g(n_i) > c$ gilt.

.

O-Kalkül

Allgemein: Es gibt eine Zahlenfolge $(n_1, n_2, \ldots)$, so dass $f(n_i)/g(n_i)$ unbegrenzt und monoton wächst
$\Rightarrow f(n) \notin O(g(n))$

Wähle zu festem $c > 0$ ein $i \in \mathbb{N}_0$ so, dass $f(n_i)/g(n_i) > c$ gilt.

Wähle zu festem $n_0 \in \mathbb{N}_0$ ein $j \geq i$ so dass gilt $n_j > n_0$.

.

Allgemein: Es gibt eine Zahlenfolge $(n_1, n_2, \ldots)$, so dass $f(n_i)/g(n_i)$ unbegrenzt und monoton wächst
$\Rightarrow f(n) \notin O(g(n))$

Wähle zu festem $c > 0$ ein $i \in \mathbb{N}_0$ so, dass $f(n_i)/g(n_i) > c$ gilt.

Wähle zu festem $n_0 \in \mathbb{N}_0$ ein $j \geq i$ so dass gilt $n_j > n_0$.

Dann gilt $cg(n_j) < (f(n_j)/g(n_j)) \cdot g(n_j) = f(n_j)$.

.

O-Kalkül

Allgemein: Es gibt eine Zahlenfolge $(n_1, n_2, \ldots)$, so dass $f(n_i)/g(n_i)$ unbegrenzt wächst $\Rightarrow f(n) \notin O(g(n))$

.

O-Kalkül

Allgemein: Es gibt eine Zahlenfolge $(n_1, n_2, \ldots)$, so dass $f(n_i)/g(n_i)$ unbegrenzt wächst $\Rightarrow f(n) \notin O(g(n))$

Entferne Elemente der Folge, für die $f(n_i)/g(n_i)$ kleiner als vorheriger Wert ist.

.