

Programmierparadigmen und Compilerbau

Syntax und Semantik
Teil 2: Auswertungsreihenfolgen

Woche 2
Sommersemester 2021
Arne Nägel

Berechnungen durch Transformationen

`quadrat x = x*x`

3 verschiedene Auswertungen für `quadrat (4+5)` :

$$\begin{array}{lcl}
 \text{quadrat}(4+5) & \xrightarrow{D} & (4+5) * (4+5) \xrightarrow{A} 9 * (4+5) \xrightarrow{A} 9 * 9 \xrightarrow{A} 81 \\
 \text{quadrat}(4+5) & \xrightarrow{D} & (4+5) * (4+5) \xrightarrow{A} (4+5) * 9 \xrightarrow{A} 9 * 9 \xrightarrow{A} 81 \\
 \text{quadrat}(4+5) & \xrightarrow{A} & \text{quadrat } 9 \xrightarrow{D} 9 * 9 \xrightarrow{A} 81
 \end{array}$$

Beobachtungen:

- Das Ergebnis ist gleich
- Die Anzahl der Reduktionen kann verschieden sein

Reduktionsstrategien

Applikative Reihenfolge:

Argumentauswertung vor Definitionseinsetzung

Normale Reihenfolge:

Definitionseinsetzung vor Argumentauswertung

Verzögerte Reihenfolge:

Normale Reihenfolge

+ Nutzung gemeinsamer Unterausdrücke (Sharing)

Einfluss der Auswertungsreihenfolge

```
quadrat (quadrat 2)
```

Argumente vor Funktionen
(Innen → außen)

```
quadrat (quadrat 2)
quadrat (2*2)
quadrat 4
4*4
16
```

„Applikative Reihenfolge“
(z.B. in C, Python, ...)

Funktionen vor Argumenten
(Außen → innen)

```
quadrat (quadrat 2)
(quadrat 2) * (quadrat 2)
(2*2) * (2*2)
4*4
16
```

„Normale Reihenfolge“
(z.B. in Haskell - in
verzögerter Variante)

Einfluss der Auswertungsreihenfolgen

```
null x = 0  
null (quadrat (quadrat 2))
```

Applikative Reihenfolge
(Innen → außen)

```
null quadrat (quadrat 2)  
null quadrat (2*2)  
null quadrat 4  
null 4*4  
null 16  
0
```

Normale Reihenfolge
(Außen → innen)

```
null (quadrat (quadrat 2))  
0
```

$$\begin{aligned} 0! &:= 1 \\ n! &:= n * (n - 1)! && \text{wenn } n > 0 \end{aligned}$$

Anwendung: $n!$ ist die Anzahl aller Permutationen der Elemente einer n -elementigen Menge.

Beispiel:

Menge mit 3 Elementen $\{A, B, C\}$:

6 Permutationen: $ABC, ACB, BAC, BCA, CAB, CBA$

$$3! = 3 * 2! = 3 * 2 * 1! = 3 * 2 * 1 * 0! = 3 * 2 * 1 * 1 = 6$$

rekursive Definition:

```
fakultaet :: Integer -> Integer
fakultaet x = if x <= 0 then 1
              else x*(fakultaet (x-1))
```

Die Funktion fakultaet ist **rekursiv**, da sie im Rumpf der Definition sich selbst referenziert

Erinnerung: Syntax

Betrachten folgende Syntax in *Backus-Naur-Form*

```

⟨definition⟩ ::= ⟨name⟩⟨name⟩* = ⟨ausdruck⟩
⟨ausdruck⟩  ::= ⟨wert⟩ | ⟨name⟩
               | (⟨ausdruck⟩ ⟨ausdruck⟩)
               | (⟨ausdruck⟩⟨infix⟩⟨ausdruck⟩)
               | if ⟨ausdruck⟩ then ⟨ausdruck⟩ else ⟨ausdruck⟩
    
```

Haben insbesondere:

- Funktionsanwendungen (D – Definitionseinsetzungen)
- Operatoranwendungen (A – Arithmetische Auswertungen)
- Bedingte Anweisungen (I – Fallunterscheidung)

Auswertung in applikativer Reihenfolge

(Argumente vor Funktionen vor; innen \rightarrow außen)

1. Ein Ausdruck der Form $\langle wert \rangle$ ist bereits ausgewertet.
2. Ein Ausdruck der Form $(\langle f \rangle \langle a_1 \rangle \dots \langle a_n \rangle)$ wird wie folgt ausgewertet:

(a) Wenn $\langle f \rangle$ kein Name ist, so werte $\langle f \rangle$ in *applikativer* Reihenfolge aus und wende das Ergebnis auf $\langle a_1 \rangle \dots \langle a_n \rangle$ an.

(b) Wenn $\langle f \rangle$ ein Name ist

i. *und nicht alle Argumentausdrücke $\langle a_1 \rangle \dots \langle a_n \rangle$ vollständig ausgewertet sind, so werte diese in applikativer Reihenfolge von links nach rechts aus.*

ii. *und alle Argumentausdrücke $\langle a_1 \rangle \dots \langle a_n \rangle$ vollständig ausgewertet sind, wende β -Reduktion an.*

Argumente
vor Funktion

3. Ein Ausdruck der Form $(\langle a_1 \rangle \langle op \rangle \langle a_2 \rangle)$, z.B. mit einem arithmetischen Operator $\langle op \rangle$, wird ausgewertet, indem zunächst $\langle a_1 \rangle$ und $\langle a_2 \rangle$ in *applikativer* Reihenfolge ausgewertet werden. Auf die resultierenden Werte wird dann der Operator $\langle op \rangle$ angewendet.

4. Ein Ausdruck der Form **if** $\langle a_b \rangle$ **then** $\langle a_t \rangle$ **else** $\langle a_e \rangle$ wird wie folgt ausgewertet:

(a) Wenn $\langle a_b \rangle$ noch nicht ausgewertet ist, so werte diesen in *applikativer* Reihenfolge aus.

(b) Ansonsten gelten die folgenden Ersetzungsregeln:

if **True** **then** $\langle a_t \rangle$ **else** $\langle a_e \rangle \rightarrow \langle a_t \rangle$

if **False** **then** $\langle a_t \rangle$ **else** $\langle a_e \rangle \rightarrow \langle a_e \rangle$

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

fakultaet 4

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

fakultaet 4

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

```
fakultaet 4
if 4 <= 1 then 1 else 4*(fakultaet (4-1))
```

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

fakultaet 4

if 4 <= 1 then 1 else 4*(fakultaet (4-1))

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

```
fakultaet 4
if 4 <= 1 then 1 else 4*(fakultaet (4-1))
if False then 1 else 4*(fakultaet (4-1))
```

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

```
fakultaet 4
if 4 <= 1 then 1 else 4*(fakultaet (4-1))
if False then 1 else 4*(fakultaet (4-1))
```



```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

```
fakultaet 4
if 4 <= 1 then 1 else 4*(fakultaet (4-1))
if False then 1 else 4*(fakultaet (4-1))
4*(fakultaet (4-1))
```

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

```
fakultaet 4
if 4 <= 1 then 1 else 4*(fakultaet (4-1))
if False then 1 else 4*(fakultaet (4-1))
4*(fakultaet (4-1))
```

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

```
fakultaet 4
if 4 <= 1 then 1 else 4*(fakultaet (4-1))
if False then 1 else 4*(fakultaet (4-1))
4*(fakultaet (4-1))
4*(fakultaet 3)
```

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

```
fakultaet 4
if 4 <= 1 then 1 else 4*(fakultaet (4-1))
if False then 1 else 4*(fakultaet (4-1))
4*(fakultaet (4-1))
4*(fakultaet 3)
```

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

```
fakultaet 4
if 4 <= 1 then 1 else 4*(fakultaet (4-1))
if False then 1 else 4*(fakultaet (4-1))
4*(fakultaet (4-1))
4*(fakultaet 3)
4*(if 3 <= 1 then 1 else 3*(fakultaet (3-1)))
```

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

applikativ

```
fakultaet 4
if 4 <= 1 then 1 else 4*(fakultaet (4-1))
if False then 1 else 4*(fakultaet (4-1))
4*(fakultaet (4-1))
4*(fakultaet 3)
4*(if 3 <= 1 then 1 else 3*(fakultaet (3-1)))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
```



```
4*(if False then 1 else 3*(fakultaet (3-1)))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))  
4*(3*(fakultaet (3-1)))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))  
4*(3*(fakultaet (3-1)))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
```

```
4*(3*(fakultaet (3-1)))
```

```
4*(3* (fakultaet 2))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
```

```
4*(3*(fakultaet (3-1)))
```

```
4*(3* (fakultaet 2))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))  
4*(3*(fakultaet (3-1)))  
4*(3* (fakultaet 2))  
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))  
4*(3*(fakultaet (3-1)))  
4*(3* (fakultaet 2))  
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))  
4*(3*(fakultaet (3-1)))  
4*(3* (fakultaet 2))  
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))  
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
```



```
4*(if False then 1 else 3*(fakultaet (3-1)))  
4*(3*(fakultaet (3-1)))  
4*(3* (fakultaet 2))  
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))  
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))  
4*(3*(fakultaet (3-1)))  
4*(3* (fakultaet 2))  
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))  
4*(3*(if False then 1 else 2*(fakultaet(2-1))))  
4*(3*(2*(fakultaet (2-1))))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))  
4*(3*(fakultaet (3-1)))  
4*(3* (fakultaet 2))  
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))  
4*(3*(if False then 1 else 2*(fakultaet(2-1))))  
4*(3*(2*(fakultaet (2-1))))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))  
4*(3*(fakultaet (3-1)))  
4*(3* (fakultaet 2))  
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))  
4*(3*(if False then 1 else 2*(fakultaet(2-1))))  
4*(3*(2*(fakultaet (2-1))))  
4*(3*(2*(fakultaet 1)))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1)))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1)))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1)))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*(if True then 1 else 1*(fakultaet(1-1)))))
```



```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*(if True then 1 else 1*(fakultaet(1-1)))))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*(if True then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*1))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*(if True then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*1))
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*(if True then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*1))
4*(3*2)
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*(if True then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*1))
4*(3*2)
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*(if True then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*1))
4*(3*2)
(4*6)
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1)))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*(if True then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*1))
4*(3*2)
(4*6)
```

```
4*(if False then 1 else 3*(fakultaet (3-1)))
4*(3*(fakultaet (3-1)))
4*(3* (fakultaet 2))
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))
4*(3*(if False then 1 else 2*(fakultaet(2-1))))
4*(3*(2*(fakultaet (2-1))))
4*(3*(2*(fakultaet 1)))
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*(if True then 1 else 1*(fakultaet(1-1)))))
4*(3*(2*1))
4*(3*2)
(4*6)
```

24


```
4*(if False then 1 else 3*(fakultaet (3-1)))  
4*(3*(fakultaet (3-1)))  
4*(3* (fakultaet 2))  
4*(3*(if 2 <= 1 then 1 else 2*(fakultaet (2-1))))  
4*(3*(if False then 1 else 2*(fakultaet(2-1))))  
4*(3*(2*(fakultaet (2-1))))  
4*(3*(2*(fakultaet 1)))  
4*(3*(2*(if 1 <= 1 then 1 else 1*(fakultaet(1-1)))))  
4*(3*(2*(if True then 1 else 1*(fakultaet(1-1)))))  
4*(3*(2*1))  
4*(3*2)  
(4*6)
```

24

18 Auswertungsschritte

```
main = const 5 (fakultaet 4)
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
const x y = x
```

$\text{main} \xrightarrow{1} \text{const } 5 \text{ (fakultaet } 4) \xrightarrow{18} \text{const } 5 \text{ 24} \xrightarrow{1} 5$

Anzahl der Reduktionen: 20

Auswertung in normaler Reihenfolge

(Funktionen vor Argumenten; außen \rightarrow innen)

1. Ein Ausdruck der Form $\langle wert \rangle$ ist bereits ausgewertet.
2. Ein Ausdruck der Form⁷ $((\langle f \rangle \langle a_1 \rangle \dots \langle a_n \rangle))$ wird wie folgt ausgewertet:
 - (a) Wenn $\langle f \rangle$ kein Name ist, so werte $\langle f \rangle$ in normaler Reihenfolge aus und wende das Ergebnis auf $\langle a_1 \rangle \dots \langle a_n \rangle$ an.
 - (b) Wenn $\langle f \rangle$ ein Name ist, wende β -Reduktion an.
3. Ein Ausdruck der Form $(\langle a_1 \rangle \langle op \rangle \langle a_2 \rangle)$, z.B. mit einem arithmetischen Operator $\langle op \rangle$, wird ausgewertet, indem zunächst $\langle a_1 \rangle$ und $\langle a_2 \rangle$ in normaler Reihenfolge ausgewertet werden. Auf die resultierenden Werte wird dann der Operator $\langle op \rangle$ angewendet.
4. Ein Ausdruck der Form **if** $\langle a_b \rangle$ **then** $\langle a_t \rangle$ **else** $\langle a_e \rangle$ wird wie folgt ausgewertet:
 - (a) Wenn $\langle a_b \rangle$ noch nicht ausgewertet ist, so werte diesen in normaler Reihenfolge aus.
 - (b) Ansonsten gelten die folgenden Ersetzungsregeln⁸:

$$\begin{array}{ll} \text{if True then } \langle a_t \rangle \text{ else } \langle a_e \rangle & \rightarrow \langle a_t \rangle \\ \text{if False then } \langle a_t \rangle \text{ else } \langle a_e \rangle & \rightarrow \langle a_e \rangle \end{array}$$

Funktion vor
Argumenten

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

Auswertung (in normaler Reihenfolge:)

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

Auswertung (in normaler Reihenfolge:)

fakultaet 4

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

Auswertung (in normaler Reihenfolge:)

fakultaet 4

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

Auswertung (in normaler Reihenfolge:)

fakultaet 4

if 4 <= 1 then 1 else 4*(fakultaet(4-1))

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

Auswertung (in normaler Reihenfolge:)

fakultaet 4

if 4 <= 1 then 1 else 4*(fakultaet(4-1))


```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

Auswertung (in normaler Reihenfolge:)

```
fakultaet 4
```

```
if 4 <= 1 then 1 else 4*(fakultaet(4-1))
```

```
if False then 1 else 4*(fakultaet(4-1))
```

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

Auswertung (in normaler Reihenfolge:)

fakultaet 4

if 4 <= 1 then 1 else 4*(fakultaet(4-1))

if False then 1 else 4*(fakultaet(4-1))

```
main = fakultaet 4
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
```

Auswertung (in normaler Reihenfolge:)

```
fakultaet 4
if 4 <= 1 then 1 else 4*(fakultaet(4-1))
if False then 1 else 4*(fakultaet(4-1))
4*(fakultaet(4-1))
```

$4 * (\text{fakultaet}(4-1)))$

4*(fakultaet(4-1))

```
4*(fakultaet(4-1)))
```

```
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
```

```
4*(fakultaet(4-1)))
```

```
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
```



```
4*(fakultaet(4-1)))
```

```
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
```

```
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(4-1)*(fakultaet((4-1)-1)))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))  
4*(3*(fakultaet((4-1)-1)))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))  
4*(3*(fakultaet((4-1)-1)))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))  
4*(3*(fakultaet((4-1)-1)))  
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))
```



```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))  
4*(3*(fakultaet((4-1)-1)))  
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))  
4*(3*(fakultaet((4-1)-1)))  
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))  
4*(3*(if(3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))  
4*(3*(fakultaet((4-1)-1)))  
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))  
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))  
4*(3*(fakultaet((4-1)-1)))  
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))  
4*(3*(if(3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))  
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))  
4*(3*(fakultaet((4-1)-1)))  
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))  
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))  
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))
4*((4-1)*(fakultaet((4-1)-1)))
4*(3*(fakultaet((4-1)-1)))
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if False then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))  
4*(3*(fakultaet((4-1)-1)))  
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))  
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))  
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))  
4*(3*(if False then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))
4*((4-1)*(fakultaet((4-1)-1)))
4*(3*(fakultaet((4-1)-1)))
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if False then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(((4-1)-1)*(fakultaet(((4-1)-1)-1))))
```



```
4*(fakultaet(4-1)))  
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))  
4*((4-1)*(fakultaet((4-1)-1)))  
4*(3*(fakultaet((4-1)-1)))  
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))  
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))  
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))  
4*(3*(if False then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))  
4*(3*((4-1)-1)*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))
4*((4-1)*(fakultaet((4-1)-1)))
4*(3*(fakultaet((4-1)-1)))
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if False then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*((3-1)*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))
4*((4-1)*(fakultaet((4-1)-1)))
4*(3*(fakultaet((4-1)-1)))
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if False then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*((3-1)*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))
4*((4-1)*(fakultaet((4-1)-1)))
4*(3*(fakultaet((4-1)-1)))
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if False then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*((3-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(2*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))
4*((4-1)*(fakultaet((4-1)-1)))
4*(3*(fakultaet((4-1)-1)))
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if False then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*((3-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(2*(fakultaet(((4-1)-1)-1))))
```

```
4*(fakultaet(4-1)))
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))
4*((4-1)*(fakultaet((4-1)-1)))
4*(3*(fakultaet((4-1)-1)))
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if False then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*((3-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(2*(fakultaet(((4-1)-1)-1))))
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))
```

```
4*(fakultaet(4-1)))
4*(if (4-1)<= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if 3 <= 1 then 1 else (4-1)*(fakultaet((4-1)-1)))
4*(if False then 1 else (4-1)*(fakultaet((4-1)-1)))
4*((4-1)*(fakultaet((4-1)-1)))
4*(3*(fakultaet((4-1)-1)))
4*(3*(if ((4-1)-1) <= 1 then 1 else (((4-1)-1)*(fakultaet(((4-1)-1)-1)))))
4*(3*(if (3-1)<= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if 2 <= 1 then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(if False then 1 else ((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(((4-1)-1)*(fakultaet(((4-1)-1)-1))))
4*(3*((3-1)*(fakultaet(((4-1)-1)-1))))
4*(3*(2*(fakultaet(((4-1)-1)-1))))
4*(3*(2*(if (((4-1)-1) <= 1 then 1 ...)))
```

`4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))`

$4 * (3 * (2 * (\text{if } ((4 - 1) - 1) \leq 1 \text{ then } 1 \dots)))$

`4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))`

`4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))`

`4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))`

`4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))`

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))  
4*(3*(2*(if True then 1 ...)))
```



```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))  
4*(3*(2*(if True then 1 ...)))
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))  
4*(3*(2*(if True then 1 ...)))  
4*(3*(2*1))
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))  
4*(3*(2*(if True then 1 ...)))  
4*(3*(2*1))
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))  
4*(3*(2*(if True then 1 ...)))  
4*(3*(2*1))  
4*(3*2)
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))  
4*(3*(2*(if True then 1 ...)))  
4*(3*(2*1))  
4*(3*2)
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))  
4*(3*(2*(if True then 1 ...)))  
4*(3*(2*1))  
4*(3*2)  
4*6
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))  
4*(3*(2*(if True then 1 ...)))  
4*(3*(2*1))  
4*(3*2)  
4*6
```

```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))  
4*(3*(2*(if True then 1 ...)))  
4*(3*(2*1))  
4*(3*2)  
4*6
```

24


```
4*(3*(2*(if (((4-1)-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if ((3-1)-1) <= 1 then 1 ...)))  
4*(3*(2*(if 2-1 <= 1 then 1 ...)))  
4*(3*(2*(if 1 <= 1 then 1 ...)))  
4*(3*(2*(if True then 1 ...)))  
4*(3*(2*1))  
4*(3*2)  
4*6
```

24

Das sind 24 Auswertungsschritte

```
main = const 5 (fakultaet 4)
fakultaet x = if x <= 1 then 1
              else x*(fakultaet (x-1))
const x y = x
```

main $\xrightarrow{1}$ const 5 (fakultaet 4) $\xrightarrow{1}$ 5

Anzahl der Reduktionen: 2

(20 bei applikativer Reihenfolge)

	verzögerte R.	applikative R.	normale R.
(fakultaet 4)	18	18	24
main	2	20	2

```
main = const 5 (fakultaet 4)
```

Es gilt: verzögerte Reduktion
hat **optimale Anzahl** von Reduktionsschritten. !

- Es gilt immer:

$$\# \text{ verzögerte R.} \leq \# \text{ normale R.}$$

$$\# \text{ verzögerte R.} \leq \# \text{ applikative R.}$$

- Im allgemeinen gilt:
 $\#$ applikative R. und $\#$ normale R. sind **unvergleichbar**
- Wenn alle Reduktionsschritte für das Ergebnis benötigt werden:

$$\# \text{ verzögerte R.} \leq \# \text{ applikative R.} \leq \# \text{ normale R.}$$