LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN



# Zentralübung 04.07.2019: Primitiv rekursive, $\mu$-rekursive Funktionen, Reduktion

Prof. Dr. David Sabel

LFE Theoretische Informatik

Letzte Änderung der Folien: 5. Juli 2019

# Wiederholung: Primitiv rekursive Funktion

## Definition

Eine Funktion $f : \mathbb{N}^k \to \mathbb{N}$ ist **primitiv rekursiv**, wenn sie der folgenden induktiven Definition genügt:

- Jede konstante Funktion $f(x_1, \ldots, x_k) = c \in \mathbb{N}$ ist primitiv rekursiv.
- Die Projektionsfunktionen $\pi_i^k(x_1, \ldots, x_k) = x_i$ sind primitiv rekursiv.
- Die Nachfolgerfunktion $succ(x) = x + 1$ ist primitiv rekursiv.
- Komposition / Einsetzung: Wenn $g : \mathbb{N}^m \to \mathbb{N}$ und für $i = 1, \ldots, m$: $h_i : \mathbb{N}^k \to \mathbb{N}$ primitiv rekursiv sind, dann ist auch $f$ mit $f(x_1, \ldots, x_k) = g(h_1(x_1, \ldots, x_k), \ldots, h_m(x_1, \ldots, x_k))$ primitiv rekursiv.
- Primitive Rekursion: Wenn $g : \mathbb{N}^{k-1} \to \mathbb{N}$ und $h : \mathbb{N}^{k+1} \to \mathbb{N}$ primitiv rekursiv, dann ist auch $f$ mit

  $$f(x_1, \ldots, x_k) = \begin{cases} g(x_2, \ldots, x_k), & \text{wenn } x_1 = 0 \\ h(f(x_1 - 1, x_2, \ldots, x_k), x_1 - 1, x_2, \ldots, x_k), & \text{sonst} \end{cases}$$

  primitiv rekursiv.

# Wiederholung: Wir wissen bereits

- $add(x, y) = x + y$ ist primitiv rekursiv
- $mult(x, y) = x * y$ ist primitiv rekursiv
- $sub(x, y) = \begin{cases} x - y, & \text{wenn } x \geq y \\ 0, & \text{sonst} \end{cases}$
  ist primitiv rekursiv
- Vertauschen, Verdoppeln, Entfernen von Argumenten ist primitiv rekursiv
- Rekursionsabstieg muss nicht über das erste, sondern kann auch über das $i$-te Argument erfolgen

## Aufgabe

Zeige, dass

$$max(x, y) = \begin{cases} x, & \text{falls } x \geq y \\ y, & \text{sonst} \end{cases}$$

primitiv rekursiv ist.

**Aufgabe**

Zeige, dass

$$max(x, y) = \begin{cases} x, & \text{falls } x \geq y \\ y, & \text{sonst} \end{cases}$$

primitiv rekursiv ist.

Es gilt $max(x, y) = add(x, sub(y, x))$, denn:

- Wenn $x \geq y$, dann $sub(y, x) = 0$ und $add(x, 0) = 0$
- Wenn $x < y$, dann $sub(y, x) = y - x$ und $add(x, y - x) = y$
- Da $add, sub$ und Komposition primitiv rekursiv, ist max auch primitiv rekursiv.

## Aufgabe

Zeige, dass

$$min(x,y) = \begin{cases} x, & \text{falls } x \leq y \\ y, & \text{sonst} \end{cases}$$

primitiv rekursiv ist.

## Aufgabe

Zeige, dass

$$min(x,y) = \begin{cases} x, & \text{falls } x \leq y \\ y, & \text{sonst} \end{cases}$$

primitiv rekursiv ist.

Beachte:

| | $sub(x,y)$ | $sub(y,x)$ | $min(x,y)$ |
|---|---|---|---|
| $x \leq y$ | $0$ | $y - x$ | $x$ |
| $x > y$ | $x - y$ | $0$ | $y$ |

## Aufgabe

Zeige, dass

$$min(x,y) = \begin{cases} x, & \text{falls } x \leq y \\ y, & \text{sonst} \end{cases}$$

primitiv rekursiv ist.

Beachte:

| | $sub(x,y)$ | $sub(y,x)$ | $min(x,y)$ |
|---|---|---|---|
| $x \leq y$ | $0$ | $y - x$ | $x{=}x{-}0{=}x{-}sub(x,y){=}sub(x,sub(x,y))$ |
| $x > y$ | $x - y$ | $0$ | $y$ |

## Aufgabe

Zeige, dass

$$min(x,y) = \begin{cases} x, & \text{falls } x \leq y \\ y, & \text{sonst} \end{cases}$$

primitiv rekursiv ist.

Beachte:

| | $sub(x,y)$ | $sub(y,x)$ | $min(x,y)$ |
|---|---|---|---|
| $x \leq y$ | $0$ | $y - x$ | $x{=}x{-}0{=}x{-}sub(x,y){=}sub(x,sub(x,y))$ |
| $x > y$ | $x - y$ | $0$ | $y{=}x{-}x{+}y{=}x{-}(x{-}y){=}sub(x,sub(x,y))$ |

## Aufgabe

Zeige, dass

$$min(x,y) = \begin{cases} x, & \text{falls } x \le y \\ y, & \text{sonst} \end{cases}$$

primitiv rekursiv ist.

Beachte:

| | $sub(x,y)$ | $sub(y,x)$ | $min(x,y)$ |
|---|---|---|---|
| $x \le y$ | $0$ | $y - x$ | $x{=}x{-}0{=}x{-}sub(x,y){=}sub(x,sub(x,y))$ |
| $x > y$ | $x - y$ | $0$ | $y{=}x{-}x{+}y{=}x{-}(x{-}y){=}sub(x,sub(x,y))$ |

Daher $min(x,y) = sub(x, sub(x,y))$

Da sub und Komposition primitiv rekursiv, ist auch $min$ primitiv rekursiv

## Aufgabe

Zeige, dass

$$absdiff(x, y) = |x - y|$$

primitiv rekursiv ist.

## Aufgabe

Zeige, dass

$$absdiff(x, y) = |x - y|$$

primitiv rekursiv ist.

Beachte:

| | $sub(x, y)$ | $sub(y, x)$ | $absdiff(x, y)$ |
|---|---|---|---|
| $x \leq y$ | $0$ | $y - x$ | $y - x$ |
| $x > y$ | $x - y$ | $0$ | $x - y$ |

## Aufgabe

Zeige, dass

$$absdiff(x, y) = |x - y|$$

primitiv rekursiv ist.

Beachte:

| | $sub(x, y)$ | $sub(y, x)$ | $absdiff(x, y)$ |
|---|---|---|---|
| $x \leq y$ | $0$ | $y - x$ | $y - x$ |
| $x > y$ | $x - y$ | $0$ | $x - y$ |

## Aufgabe

Zeige, dass

$$absdiff(x, y) = |x - y|$$

primitiv rekursiv ist.

Beachte:

| | $sub(x, y)$ | $sub(y, x)$ | $absdiff(x, y)$ |
|---|---|---|---|
| $x \leq y$ | $0$ | $y - x$ | $y - x$ |
| $x > y$ | $x - y$ | $0$ | $x - y$ |

## Aufgabe

Zeige, dass

$$absdiff(x, y) = |x - y|$$

primitiv rekursiv ist.

Beachte:

| | $sub(x, y)$ | $sub(y, x)$ | $absdiff(x, y)$ |
|---|---|---|---|
| $x \leq y$ | $0$ | $y - x$ | $y - x$ |
| $x > y$ | $x - y$ | $0$ | $x - y$ |

Daher $absdiff(x, y) = add(sub(x, y), sub(y, x))$

Da sub,add und Komposition primitiv rekursiv, ist auch $absdiff$ primitiv rekursiv

### Aufgabe

Zeige, dass die Fakultätsfunktion mit $fac(0) = 1$ und $fac(n) = \prod_{i=1}^{n} i$ für $n > 0$ primitiv-rekursiv ist.

Primitiv-rekursive Definition:

$$fac(x) = \begin{cases} g() & \text{falls } x = 0 \\ h(fac(x-1), x-1) & \text{sonst} \end{cases}$$

wobei:

- $g() =?$
- $h(x, y) =?$

## Aufgabe

Zeige, dass die Fakultätsfunktion mit $fac(0) = 1$ und $fac(n) = \prod_{i=1}^{n} i$ für $n > 0$ primitiv-rekursiv ist.

Primitiv-rekursive Definition:

$$fac(x) = \begin{cases} g() & \text{falls } x = 0 \\ h(fac(x-1), x-1) & \text{sonst} \end{cases}$$

wobei:

- $g() = 1$
- $h(x, y) =?$

## Aufgabe

Zeige, dass die Fakultätsfunktion mit $fac(0) = 1$ und $fac(n) = \prod_{i=1}^{n} i$ für $n > 0$ primitiv-rekursiv ist.

Primitiv-rekursive Definition:

$$fac(x) = \begin{cases} g() & \text{falls } x = 0 \\ h(fac(x-1), x-1) & \text{sonst} \end{cases}$$

wobei:

- $g() = 1$
- $h(x, y) = mult(x, add(y, 1))$

## Aufgabe

Zeige, dass die Fakultätsfunktion mit $fac(0) = 1$ und $fac(n) = \prod_{i=1}^{n} i$ für $n > 0$ primitiv-rekursiv ist.

Primitiv-rekursive Definition:

$$fac(x) = \begin{cases} g() & \text{falls } x = 0 \\ h(fac(x-1), x-1) & \text{sonst} \end{cases}$$

wobei:

- $g() = 1$
- $h(x, y) = mult(x, add(y, 1)) = mult(\pi_1^2(x, y), r(x, y))$ mit
- $r(a, b) = add(\pi_2^2(a, b), const_1(a, b))$
- $const_1(a, b) = 1$

# Primitiv rekursive Prädikate

- Primitiv rekursive Funktionen $\mathbb{N}^k \to \mathbb{N}$
- Prädikate liefern eigentlich wahr oder falsch.
- Wir verwenden $P : \mathbb{N}^k \to \{0, 1\}$ für Prädikate
- Das passt immer noch zu den primitiv rekursiven Funktionen

**Aufgabe**

Zeige, dass das Prädikat

$$equal(x, y) = \begin{cases} 1, & \text{wenn } x = y \\ 0, & \text{sonst} \end{cases}$$

primitiv rekursiv ist.

Beobachtung: $equal(x, y)$ gilt, g.d.w. $|x - y| = 0$

- $equal(x, y) = eq0?(absdiff(x, y))$
- wobei $eq0?(x) = \begin{cases} 1, & \text{wenn } x = 0 \\ 0, & \text{sonst} \end{cases}$
- $eq0?(x) = \begin{cases} g(), & \text{wenn } x = 0 \\ h(eq0(x-1), x-1), & \text{wenn } x > 0 \end{cases}$

  mit $g() = 1$ und $h(x, y) = 0$

Der beschränkte maximum-Operator $\overline{max}_P : \mathbb{N}^k \to \mathbb{N}$ für ein $k$-stelliges Prädikat $P : \mathbb{N}^k \to \{0,1\}$ sei definiert durch

$$\overline{max}_P(n, x_2, \dots, x_k) := \begin{cases} \max\{x_1 \leq n \mid P(x_1, x_2, \dots, x_k)\}, & \text{wenn es ein } 0 \leq x_1 \leq n \text{ gibt} \\ & \text{mit } P(x_1, x_2, \dots, x_k) = 1. \\ 0, & \text{sonst} \end{cases}$$

D.h. der $\overline{max}_P(n, x_2, \dots, x_k)$-Operator sucht nach einem maximalen Wert $x_1$, sodass $P(x_1, \dots, x_k)$ erfüllt ist und $x_1$ nicht größer als $n$ ist.

**Aufgabe**

Zeige: Wenn $P$ primitiv rekursiv ist, dann ist auch $\overline{max}_P$ primitiv rekursiv

# Beschränkter maximum-Operator

$$\overline{max}_P(x_1,\ldots,x_k) = \begin{cases} 0, & \text{wenn } x_1 = 0 \\ x_1, & \text{wenn } P(x_1,\ldots,x_k) = 1 \\ \overline{max}_P(x_1 - 1, x_2, \ldots, x_k), & \text{sonst} \end{cases}$$

Äquivalent dazu ist die Definition:

$$\overline{max}_P(x_1,\ldots,x_k) = \begin{cases} 0, & \text{wenn } x_1 = 0 \\ \overline{max}_P(x_1 - 1, x_2, \ldots, x_k) & \\ +P(x_1,\ldots,x_k) & \\ \quad * (x_1 - \overline{max}_P(x_1 - 1, x_2, \ldots, x_k)), & \text{sonst} \end{cases}$$

Diese Definition passt zur primitiven Rekursion mit

$$\begin{aligned} g(x_2,\ldots,x_k) &= 0 \\ h(x_0, x_1, \ldots, x_k) &= add(x_0, mult(P(x_1,\ldots,x_j), (sub(x_1, x_0)))) \end{aligned}$$

# Wiederholung: $\mu$-Rekursion

**Definition $\mu$-Operator**

Sei $h : \mathbb{N}^{k+1} \to \mathbb{N}$ eine (partielle oder totale) Funktion. Dann ist $(\mu h) : \mathbb{N}^k \to \mathbb{N}$ definiert als

$$(\mu h)(x_1, \ldots, x_k) = \begin{cases} n & \text{falls } h(n, x_1, \ldots, x_k) = 0 \text{ und für} \\ & \text{alle } m < n\text{: } h(m, x_1, \ldots, x_k) \text{ ist} \\ & \text{definiert und } h(m, x_1, \ldots, x_k) > 0. \\ \text{undefiniert,} & \text{sonst} \end{cases}$$

- $\mu$-Operator „sucht“ nach einer kleinsten Nullstelle von $h$.
- Wenn diese nicht existiert (entweder da $h$ keine Nullstelle hat, oder da $h$ undefiniert ist für Werte, die kleiner als die Nullstelle sind), dann ist auch der $\mu$-Operator angewendet auf $h$ undefiniert.

# $\mu$-rekursive Funktionen

## Definition

Die Menge aller $\mu$-rekursiven Funktionen sei die kleinste Menge, so dass gilt:

- Jede konstante Funktion $f(x_1, \ldots, x_k) = c \in \mathbb{N}$ ist $\mu$-rekursiv.
- Die Projektionsfunktionen $\pi_i^k(x_1, \ldots, x_k) = x_i$ sind $\mu$-rekursiv.
- Die Nachfolgerfunktion $succ(x) = x + 1$ ist $\mu$-rekursiv.
- Komposition / Einsetzung: Wenn $g : \mathbb{N}^m \to \mathbb{N}$ und für $i = 1, \ldots, m$: $h_i : \mathbb{N}^k \to \mathbb{N}$ $\mu$-rekursiv sind, dann ist auch $f$ mit $f(x_1, \ldots, x_k) = g(h_1(x_1, \ldots, x_k), \ldots, h_m(x_1, \ldots, x_k))$ $\mu$-rekursiv.
- Rekursion: Wenn $g : \mathbb{N}^{k-1} \to \mathbb{N}$ und $h : \mathbb{N}^{k+1} \to \mathbb{N}$ $\mu$-rekursiv, dann ist

  $$f(x_1, \ldots, x_k) = \begin{cases} g(x_2, \ldots, x_k), & \text{wenn } x_1 = 0 \\ h(f(x_1 - 1, x_2, \ldots, x_k), x_1 - 1, x_2, \ldots, x_k), & \text{sonst} \end{cases}$$

  auch $\mu$-rekursiv.
- $\mu$-Operator: Wenn $h : \mathbb{N}^{k+1} \to \mathbb{N}$ $\mu$-rekursiv, dann auch $f = \mu h$ $\mu$-rekursiv.

## Aufgabe

Welche Funktion wird durch $(\mu h_i)$ mit

a) $h_1(x, y) = sub(x, y)$

b) $h_2(x, y) = sub(y, x)$

c) $h_3(x, y) = sub(y, mult(2, x))$

d) $h_4(x, y) = sub(y, mult(x, x))$

e) $h_5(x, y) = min(x, y)$

f) $h_6(x, y) = max(x, y)$

berechnet?

a) $h_1(x, y) = sub(x, y)$

a) $h_1(x, y) = sub(x, y)$

- $(\mu h_1)$ berechnet kleinestes $x$, sodass $sub(x, y) = 0$

a) $h_1(x, y) = sub(x, y)$

- $(\mu h_1)$ berechnet kleinestes $x$, sodass $sub(x, y) = 0$
- das gilt schon für $x = 0$ also $\mu h_1(y) = 0$

a) $h_1(x, y) = sub(x, y)$

- $(\mu h_1)$ berechnet kleinestes $x$, sodass $sub(x, y) = 0$
- das gilt schon für $x = 0$ also $\mu h_1(y) = 0$

b) $h_2(x, y) = sub(y, x)$

- $(\mu h_2)$ berechnet kleinstes $x$, sodass $sub(y, x) = 0$

a) $h_1(x, y) = sub(x, y)$

- $(\mu h_1)$ berechnet kleinestes $x$, sodass $sub(x, y) = 0$
- das gilt schon für $x = 0$ also $\mu h_1(y) = 0$

b) $h_2(x, y) = sub(y, x)$

- $(\mu h_2)$ berechnet kleinstes $x$, sodass $sub(y, x) = 0$
- gilt für $x = y$

a) $h_1(x, y) = sub(x, y)$

- $(\mu h_1)$ berechnet kleinestes $x$, sodass $sub(x, y) = 0$
- das gilt schon für $x = 0$ also $\mu h_1(y) = 0$

b) $h_2(x, y) = sub(y, x)$

- $(\mu h_2)$ berechnet kleinstes $x$, sodass $sub(y, x) = 0$
- gilt für $x = y$
- für alle $m < y$ gilt $sub(y, m) > 0$

a) $h_1(x, y) = sub(x, y)$

- $(\mu h_1)$ berechnet kleinestes $x$, sodass $sub(x, y) = 0$
- das gilt schon für $x = 0$ also $\mu h_1(y) = 0$

b) $h_2(x, y) = sub(y, x)$

- $(\mu h_2)$ berechnet kleinstes $x$, sodass $sub(y, x) = 0$
- gilt für $x = y$
- für alle $m < y$ gilt $sub(y, m) > 0$
- daher $(\mu h_2)(y) = y$

c) $h_3(x, y) = sub(y, mult(2, x))$

c) $h_3(x, y) = sub(y, mult(2, x))$

- finde kleinstes $x$ sodass $y - 2x \leq 0$

c) $h_3(x, y) = sub(y, mult(2, x))$

- finde kleinstes $x$ sodass $y - 2x \leq 0$
- gilt für $x = \lceil y/2 \rceil$

c) $h_3(x, y) = sub(y, mult(2, x))$

- finde kleinstes $x$ sodass $y - 2x \leq 0$
- gilt für $x = \lceil y/2 \rceil$
- $(\mu h_3)(y) = \lceil y/2 \rceil$

c) $h_3(x, y) = sub(y, mult(2, x))$

- finde kleinstes $x$ sodass $y - 2x \leq 0$
- gilt für $x = \lceil y/2 \rceil$
- $(\mu h_3)(y) = \lceil y/2 \rceil$

d) $h_4(x, y) = sub(y, mult(x, x))$

c) $h_3(x, y) = sub(y, mult(2, x))$

- finde kleinstes $x$ sodass $y - 2x \leq 0$
- gilt für $x = \lceil y/2 \rceil$
- $(\mu h_3)(y) = \lceil y/2 \rceil$

d) $h_4(x, y) = sub(y, mult(x, x))$

- finde kleinstes $x$ sodass $y - x * x \leq 0$

c) $h_3(x, y) = sub(y, mult(2, x))$

- finde kleinstes $x$ sodass $y - 2x \leq 0$
- gilt für $x = \lceil y/2 \rceil$
- $(\mu h_3)(y) = \lceil y/2 \rceil$

d) $h_4(x, y) = sub(y, mult(x, x))$

- finde kleinstes $x$ sodass $y - x * x \leq 0$
- gilt für $x = \lceil \sqrt{y} \rceil$

c) $h_3(x, y) = sub(y, mult(2, x))$

- finde kleinstes $x$ sodass $y - 2x \leq 0$
- gilt für $x = \lceil y/2 \rceil$
- $(\mu h_3)(y) = \lceil y/2 \rceil$

d) $h_4(x, y) = sub(y, mult(x, x))$

- finde kleinstes $x$ sodass $y - x * x \leq 0$
- gilt für $x = \lceil \sqrt{y} \rceil$
- $(\mu h_4)(y) = \lceil \sqrt{y} \rceil$

e) $h_5(x, y) = min(x, y)$

e) $h_5(x, y) = min(x, y)$

- finde kleinstes $x$ mit $min(x, y) = 0$

e) $h_5(x, y) = min(x, y)$

- finde kleinstes $x$ mit $min(x, y) = 0$
- gilt schon für $x = 0$

e) $h_5(x, y) = min(x, y)$

- finde kleinstes $x$ mit $min(x, y) = 0$
- gilt schon für $x = 0$
- $(\mu h_5)(y) = 0$

e) $h_5(x, y) = min(x, y)$

- finde kleinstes $x$ mit $min(x, y) = 0$
- gilt schon für $x = 0$
- $(\mu h_5)(y) = 0$

f) $h_6(x, y) = max(x, y)$

e) $h_5(x, y) = min(x, y)$

- finde kleinstes $x$ mit $min(x, y) = 0$
- gilt schon für $x = 0$
- $(\mu h_5)(y) = 0$

f) $h_6(x, y) = max(x, y)$

- finde kleinstes $x$ mit $max(x, y) = 0$

e) $h_5(x, y) = min(x, y)$

- finde kleinstes $x$ mit $min(x, y) = 0$
- gilt schon für $x = 0$
- $(\mu h_5)(y) = 0$

f) $h_6(x, y) = max(x, y)$

- finde kleinstes $x$ mit $max(x, y) = 0$
- $0$, falls $y = 0$ und ansonsten undefiniert

e) $h_5(x, y) = min(x, y)$

- finde kleinstes $x$ mit $min(x, y) = 0$
- gilt schon für $x = 0$
- $(\mu h_5)(y) = 0$

f) $h_6(x, y) = max(x, y)$

- finde kleinstes $x$ mit $max(x, y) = 0$
- $0$, falls $y = 0$ und ansonsten undefiniert
- $(\mu h_6)(y) = \begin{cases} 0, & \text{falls } y = 0 \\ \perp, & \text{sonst} \end{cases}$

# Beispiel: Reduktionen bei entscheidbaren Sprachen

**Aufgabe**

Sei $L_1 \subseteq \Sigma_1^* = \{a, b\}^*$ und $L_2 \subseteq \Sigma_2^* = \{c, d\}^*$ mit

$$L_1 = \{a^n b^n \mid n \in \mathbb{N}\} \qquad L_2 = \{c^m d^m \mid m \in \mathbb{N}\}$$

Zeige $L_1 \leq L_2$.

- Zeige: Es existiert totale, berechenbare Funktion $f : \Sigma_1^* \to \Sigma_2^*$, mit

$$\forall w \in \Sigma_1^* : w \in L_1 \iff f(w) \in L_2$$

# Beispiel: Reduktionen bei entscheidbaren Sprachen

$$L_1 = \{a^n b^n \mid n \in \mathbb{N}\} \qquad L_2 = \{c^m d^m \mid m \in \mathbb{N}\}$$

- Sei $f(aw) := cf(w)$, $f(bw) := df(w)$, $f(\varepsilon) := \varepsilon$

# Beispiel: Reduktionen bei entscheidbaren Sprachen

$$L_1 \quad = \quad \{a^n b^n \mid n \in \mathbb{N}\} \qquad L_2 \quad = \quad \{c^m d^m \mid m \in \mathbb{N}\}$$

- Sei $f(aw) := cf(w)$, $f(bw) := df(w)$, $f(\varepsilon) := \varepsilon$
- $f$ ist berechenbar, da sie von der folgenden TM berechnet wird: Die TM liest die Eingabe von links nach rechts, ersetzt alle $a$ durch $c$ und alle $b$ durch $d$ und fährt am Ende den Kopf an den Anfang.

# Beispiel: Reduktionen bei entscheidbaren Sprachen

$$L_1 \quad = \quad \{a^n b^n \mid n \in \mathbb{N}\} \qquad L_2 \quad = \quad \{c^m d^m \mid m \in \mathbb{N}\}$$

- Sei $f(aw) := cf(w)$, $f(bw) := df(w)$, $f(\varepsilon) := \varepsilon$
- $f$ ist berechenbar, da sie von der folgenden TM berechnet wird: Die TM liest die Eingabe von links nach rechts, ersetzt alle $a$ durch $c$ und alle $b$ durch $d$ und fährt am Ende den Kopf an den Anfang.
- Wenn $w = a^n b^n$, dann $f(w) = c^n d^n$

# Beispiel: Reduktionen bei entscheidbaren Sprachen

$$L_1 \;=\; \{a^n b^n \mid n \in \mathbb{N}\} \qquad L_2 \;=\; \{c^m d^m \mid m \in \mathbb{N}\}$$

- Sei $f(aw) := cf(w)$, $f(bw) := df(w)$, $f(\varepsilon) := \varepsilon$
- $f$ ist berechenbar, da sie von der folgenden TM berechnet wird: Die TM liest die Eingabe von links nach rechts, ersetzt alle $a$ durch $c$ und alle $b$ durch $d$ und fährt am Ende den Kopf an den Anfang.
- Wenn $w = a^n b^n$, dann $f(w) = c^n d^n$
- Wenn $w \neq a^n b^n$, dann $f(w) \neq c^m d^m$ für irgendein $m$.

# Beispiel: Reduktionen bei entscheidbaren Sprachen

$$L_1 \quad = \quad \{a^n b^n \mid n \in \mathbb{N}\} \qquad L_2 \quad = \quad \{c^m d^m \mid m \in \mathbb{N}\}$$

- Sei $f(aw) := cf(w)$, $f(bw) := df(w)$, $f(\varepsilon) := \varepsilon$
- $f$ ist berechenbar, da sie von der folgenden TM berechnet wird: Die TM liest die Eingabe von links nach rechts, ersetzt alle $a$ durch $c$ und alle $b$ durch $d$ und fährt am Ende den Kopf an den Anfang.
- Wenn $w = a^n b^n$, dann $f(w) = c^n d^n$
- Wenn $w \neq a^n b^n$, dann $f(w) \neq c^m d^m$ für irgendein $m$.
- Daher gilt für alle $w \in \Sigma_1^* : w \in L_1 \iff f(w) \in L_2$

# Beispiel: Reduktionen bei entscheidbaren Sprachen

$$L_1 \;=\; \{a^n b^n \mid n \in \mathbb{N}\} \qquad L_2 \;=\; \{c^m d^m \mid m \in \mathbb{N}\}$$

- Sei $f(aw) := cf(w)$, $f(bw) := df(w)$, $f(\varepsilon) := \varepsilon$
- $f$ ist berechenbar, da sie von der folgenden TM berechnet wird: Die TM liest die Eingabe von links nach rechts, ersetzt alle $a$ durch $c$ und alle $b$ durch $d$ und fährt am Ende den Kopf an den Anfang.
- Wenn $w = a^n b^n$, dann $f(w) = c^n d^n$
- Wenn $w \neq a^n b^n$, dann $f(w) \neq c^m d^m$ für irgendein $m$.
- Daher gilt für alle $w \in \Sigma_1^* : w \in L_1 \iff f(w) \in L_2$
- und daher $L_1 \leq L_2$

# Beispiel: Reduktionen bei entscheidbaren Sprachen

$$L_1 \;=\; \{a^n b^n \mid n \in \mathbb{N}\} \qquad L_2 \;=\; \{c^m d^m \mid m \in \mathbb{N}\}$$

- Sei $f(aw) := cf(w)$, $f(bw) := df(w)$, $f(\varepsilon) := \varepsilon$
- $f$ ist berechenbar, da sie von der folgenden TM berechnet wird: Die TM liest die Eingabe von links nach rechts, ersetzt alle $a$ durch $c$ und alle $b$ durch $d$ und fährt am Ende den Kopf an den Anfang.
- Wenn $w = a^n b^n$, dann $f(w) = c^n d^n$
- Wenn $w \neq a^n b^n$, dann $f(w) \neq c^m d^m$ für irgendein $m$.
- Daher gilt für alle $w \in \Sigma_1^* : w \in L_1 \iff f(w) \in L_2$
- und daher $L_1 \leq L_2$

Damit folgt:

- Wenn $L_2$ entscheidbar, dann auch $L_1$ entscheidbar.

# Beispiel: Reduktionen bei entscheidbaren Sprachen

$$L_1 = \{a^n b^n \mid n \in \mathbb{N}\} \qquad L_2 = \{c^m d^m \mid m \in \mathbb{N}\}$$

- Sei $f(aw) := cf(w)$, $f(bw) := df(w)$, $f(\varepsilon) := \varepsilon$
- $f$ ist berechenbar, da sie von der folgenden TM berechnet wird: Die TM liest die Eingabe von links nach rechts, ersetzt alle $a$ durch $c$ und alle $b$ durch $d$ und fährt am Ende den Kopf an den Anfang.
- Wenn $w = a^n b^n$, dann $f(w) = c^n d^n$
- Wenn $w \neq a^n b^n$, dann $f(w) \neq c^m d^m$ für irgendein $m$.
- Daher gilt für alle $w \in \Sigma_1^* : w \in L_1 \iff f(w) \in L_2$
- und daher $L_1 \leq L_2$

Damit folgt:

- Wenn $L_2$ entscheidbar, dann auch $L_1$ entscheidbar.
- Wenn $L_1$ unentscheidbar, dann auch $L_2$ unentscheidbar.

# Beispiel: Reduktionen bei entscheidbaren Sprachen

Die Funktion

$$f(w) = \begin{cases} cd, & \text{wenn } w = a^n b^n \\ ccd, & \text{sonst} \end{cases}$$

zeigt auch $L_1 \leq L_2$, denn $w \in L_1 \iff f(w) \in L_2$
allerdings ist es hier schwieriger zu zeigen, das $f$ berechenbar ist!

Dafür muss man in diesem Fall schon zeigen, dass $L_1$ entscheidbar ist!