Zentralübung 6

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 3. Juli 2024
Basierend auf Folien von PD Dr. David Sabel

Plan für heute

1. Reduktionen
2. Der Satz von Rice
3. Das Postsche Korrespondenzproblem (PCP)
4. Primitiv und μ-rekursive Funktionen (nur FSK)

1. Reduktionen

Reduktion

Definition

Seien $L_1 \subseteq \Sigma_1^*$ und $L_2 \subseteq \Sigma_2^*$ Sprachen.
Dann sagen wir L_1 ist auf L_2 reduzierbar (geschrieben $L_1 \leq L_2$), falls es eine totale berechenbare Funktion $f : \Sigma_1^* \to \Sigma_2^*$ gibt, sodass für alle $w \in \Sigma_1^*$ gilt: $w \in L_1$ g.d.w. $f(w) \in L_2$.
Die Funktion f nennt man Reduktion.

Eselsbrücke:

$$\underbrace{L_1}_{\text{„kleines" Problem}} \leq \underbrace{L_2}_{\text{„großes" Problem}}$$

- $\leq$ sagt die Wahrheit.
- „Reduktion“ täuscht.
  Man reduziert das „kleine“ Problem auf das „große“.

Aufgabe: Reduktion

Zeige $L = \{w \mid \text{TM } M_w \text{ berechnet 1 bei Eingabe 0}\}$ ist unentscheidbar.

Aufgabe: Reduktion

Zeige $L = \{w \mid \text{TM } M_w \text{ berechnet } 1 \text{ bei Eingabe } 0\}$ ist unentscheidbar.

Antwort:

Aufgabe: Reduktion

Zeige $L = \{w \mid \text{TM } M_w \text{ berechnet } 1 \text{ bei Eingabe } 0\}$ ist unentscheidbar.

Antwort:

1. Wähle unentscheidbare Sprache, die wir auf L reduzieren:

Aufgabe: Reduktion

Zeige $L = \{w \mid \text{TM } M_w \text{ berechnet 1 bei Eingabe 0}\}$ ist unentscheidbar.

Antwort:

1. Wähle unentscheidbare Sprache, die wir auf L reduzieren:

   Wir reduzieren H_0 auf L (d.h. $H_0 \leq L$).
   $H_0 = \{w \mid \text{TM } M_w \text{ hält bei leerer Eingabe}\}$.

Aufgabe: Reduktion

Zeige $L = \{w \mid \text{TM } M_w \text{ berechnet 1 bei Eingabe 0}\}$ ist unentscheidbar.

Antwort:

1. Wähle unentscheidbare Sprache, die wir auf L reduzieren:

   Wir reduzieren H_0 auf L (d.h. $H_0 \leq L$).
   $H_0 = \{w \mid \text{TM } M_w \text{ hält bei leerer Eingabe}\}$.
2. Definiere f (mit $w \in H_0$ g.d.w. $f(w) \in L$):

Aufgabe: Reduktion

Zeige $L = \{w \mid \text{TM } M_w \text{ berechnet 1 bei Eingabe 0}\}$ ist unentscheidbar.

Antwort:

1. Wähle unentscheidbare Sprache, die wir auf L reduzieren:

   Wir reduzieren H_0 auf L (d.h. $H_0 \leq L$).
   $H_0 = \{w \mid \text{TM } M_w \text{ hält bei leerer Eingabe}\}$.
2. Definiere f (mit $w \in H_0$ g.d.w. $f(w) \in L$):

   Sei f die folgende Funktion, die bei Eingabe w, eine neue Turingmaschinenbeschreibung u erstellt, sodass M_u sich wie folgt verhält:
   - M_u löscht das Band und simuliert M_w auf leerer Eingabe.
   - Wenn M_w anhält, dann schreibe 1 auf das Band und akzeptiere.

Aufgabe: Reduktion

Zeige $L = \{w \mid \text{TM } M_w \text{ berechnet 1 bei Eingabe 0}\}$ ist unentscheidbar.

Antwort:

1. Wähle unentscheidbare Sprache, die wir auf L reduzieren:

   Wir reduzieren H_0 auf L (d.h. $H_0 \leq L$).
   $H_0 = \{w \mid \text{TM } M_w \text{ hält bei leerer Eingabe}\}$.

2. Definiere f (mit $w \in H_0$ g.d.w. $f(w) \in L$):

   Sei f die folgende Funktion, die bei Eingabe w, eine neue Turingmaschinenbeschreibung u erstellt, sodass M_u sich wie folgt verhält:

   - M_u löscht das Band und simuliert M_w auf leerer Eingabe.
   - Wenn M_w anhält, dann schreibe 1 auf das Band und akzeptiere.

3. Zeige, dass f total und berechenbar ist:

Aufgabe: Reduktion

Zeige $L = \{w \mid \text{TM } M_w \text{ berechnet 1 bei Eingabe 0}\}$ ist unentscheidbar.

Antwort:

1. Wähle unentscheidbare Sprache, die wir auf L reduzieren:

   Wir reduzieren H_0 auf L (d.h. $H_0 \leq L$).
   $H_0 = \{w \mid \text{TM } M_w \text{ hält bei leerer Eingabe}\}$.
2. Definiere f (mit $w \in H_0$ g.d.w. $f(w) \in L$):

   Sei f die folgende Funktion, die bei Eingabe w, eine neue Turingmaschinenbeschreibung u erstellt, sodass M_u sich wie folgt verhält:
   - M_u löscht das Band und simuliert M_w auf leerer Eingabe.
   - Wenn M_w anhält, dann schreibe 1 auf das Band und akzeptiere.
3. Zeige, dass f total und berechenbar ist:

   Totalität ist klar. Die TM M_u ist konstruierbar und das (De)kodieren von w und u auf TM ist möglich.

Aufgabe: Reduktion

4. Zeige $w \in H_0$ g.d.w. $f(w) \in L$:

Aufgabe: Reduktion

4. Zeige $w \in H_0$ g.d.w. $f(w) \in L$:

   Es gilt:

   $$w \in H_0$$

Aufgabe: Reduktion

4. Zeige $w \in H_0$ g.d.w. $f(w) \in L$:

   Es gilt:

   $w \in H_0$

   g.d.w. M_w hält bei leerer Eingabe

Aufgabe: Reduktion

4. Zeige $w \in H_0$ g.d.w. $f(w) \in L$:

   Es gilt:

   $w \in H_0$

   g.d.w. M_w hält bei leerer Eingabe

   g.d.w. M_u hält bei jeder Eingabe (auch bei 0) und schreibt 1 auf das Band

Aufgabe: Reduktion

4. Zeige $w \in H_0$ g.d.w. $f(w) \in L$:

   Es gilt:

   $w \in H_0$

   g.d.w. M_w hält bei leerer Eingabe

   g.d.w. M_u hält bei jeder Eingabe (auch bei 0) und schreibt 1 auf das Band

   g.d.w. M_u berechnet 1 bei Eingabe 0

Aufgabe: Reduktion

4. Zeige $w \in H_0$ g.d.w. $f(w) \in L$:

   Es gilt:

   $w \in H_0$
   g.d.w. M_w hält bei leerer Eingabe
   g.d.w. M_u hält bei jeder Eingabe (auch bei 0) und schreibt 1 auf das Band
   g.d.w. M_u berechnet 1 bei Eingabe 0
   g.d.w. $u = f(w) \in L$

Aufgabe: Reduktion

4. Zeige $w \in H_0$ g.d.w. $f(w) \in L$:

   Es gilt:

   $w \in H_0$
   g.d.w. M_w hält bei leerer Eingabe
   g.d.w. M_u hält bei jeder Eingabe (auch bei 0) und schreibt 1 auf das Band
   g.d.w. M_u berechnet 1 bei Eingabe 0
   g.d.w. $u = f(w) \in L$

5. Damit gilt $H_0 \leq L$. Da H_0 unentscheidbar ist, ist auch L unentscheidbar.

2. Der Satz von Rice

Der Satz von Rice

Satz von Rice

Sei $\mathcal{R}$ die Klasse aller turingberechenbaren Funktionen. Sei $\mathcal{S}$ eine beliebige Teilmenge, sodass $\emptyset \subset \mathcal{S} \subset \mathcal{R}$. Dann ist folgende Sprache unentscheidbar:

$$C(\mathcal{S}) = \{w \mid \text{die von } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\}$$

Anwendung des Satzes von Rice

Sei L eine Sprache, die als unentscheidbar zu beweisen ist.

Schritte:

1. Definiere Menge S von Funktionen.
2. Zeige Nichttrivialität von S.
3. Begründe, dass S richtig gewählt ist, d.h. $C(S) = L$.
4. Der Satz von Rice zeigt dann das Resultat.

Aufgabe: Den Satz von Rice anwenden

Sei $L = \{w \mid \text{wenn } M_w \text{ auf Eingabe 0 hält, dann hält } M_w \text{ auf Eingabe 1}\}$.

Antwort:

1. $S = \{f \mid f \text{ ist an der Stelle 0 undefiniert oder an den Stellen 0 und 1 definiert}\}$.

Aufgabe: Den Satz von Rice anwenden

Sei $L = \{w \mid \text{wenn } M_w \text{ auf Eingabe 0 hält, dann hält } M_w \text{ auf Eingabe 1}\}$.

Antwort:

1. $S = \{f \mid f \text{ ist an der Stelle 0 undefiniert oder an den Stellen 0 und 1 definiert}\}$.
2. S ist nicht trivial:
   - $\emptyset \subset S$, da $id \in S$ mit $id(x) = x$.
   - $S \subset \mathcal{R}$, da für f, mit $f(0) = 1$ aber $f(i)$ undefiniert für $i \neq 0$, gilt $f \notin S$.

Aufgabe: Den Satz von Rice anwenden

Sei $L = \{w \mid \text{wenn } M_w \text{ auf Eingabe 0 hält, dann hält } M_w \text{ auf Eingabe 1}\}$.

Antwort:

1. $S = \{f \mid f \text{ ist an der Stelle 0 undefiniert oder an den Stellen 0 und 1 definiert}\}$.
2. S ist nicht trivial:
   - $\emptyset \subset S$, da $id \in S$ mit $id(x) = x$.
   - $S \subset \mathcal{R}$, da für f, mit $f(0) = 1$ aber $f(i)$ undefiniert für $i \neq 0$, gilt $f \notin S$.
3. $$\begin{aligned} C(S) &= \{w \mid \text{die von } M_w \text{ berechnete Funktion liegt in } S\} \\ &= \{w \mid \text{wenn } M_w \text{ auf Eingabe 0 hält, dann hält } M_w \text{ auf Eingabe 1}\} \\ &= L. \end{aligned}$$

Aufgabe: Den Satz von Rice anwenden

Sei $L = \{w \mid \text{wenn } M_w \text{ auf Eingabe 0 hält, dann hält } M_w \text{ auf Eingabe 1}\}$.

Antwort:

1. $S = \{f \mid f \text{ ist an der Stelle 0 undefiniert oder an den Stellen 0 und 1 definiert}\}$.
2. S ist nicht trivial:
   - $\emptyset \subset S$, da $id \in S$ mit $id(x) = x$.
   - $S \subset \mathcal{R}$, da für f, mit $f(0) = 1$ aber $f(i)$ undefiniert für $i \neq 0$, gilt $f \notin S$.
3. $$\begin{aligned} C(S) &= \{w \mid \text{die von } M_w \text{ berechnete Funktion liegt in } S\} \\ &= \{w \mid \text{wenn } M_w \text{ auf Eingabe 0 hält, dann hält } M_w \text{ auf Eingabe 1}\} \\ &= L. \end{aligned}$$
4. Mit dem Satz von Rice ist $C(S)$ unentscheidbar.

3. Das Postsche Korrespondenzproblem (PCP)

Ein Verschlüsselungsproblem

Eine Verschlüsselungstabelle ist eine Tabelle T

Wort	Kodewort
w_1	k_1
$\vdots$	$\vdots$
w_n	k_n

wobei $w_i, k_i \in \Sigma^+$.

Ein Verschlüsselungsproblem

Eine Verschlüsselungstabelle ist eine Tabelle T

Wort	Kodewort
w_1	k_1
$\vdots$	$\vdots$
w_n	k_n

wobei $w_i, k_i \in \Sigma^+$.

Kodierung von ganzen Zeichenketten: aus $w = w_{i_1} \cdots w_{i_m}$ wird $k_{i_1} \cdots k_{i_m}$.

Ein Verschlüsselungsproblem

Eine Verschlüsselungstabelle ist eine Tabelle T

Wort	Kodewort
w_1	k_1
$\vdots$	$\vdots$
w_n	k_n

wobei $w_i, k_i \in \Sigma^+$.

Kodierung von ganzen Zeichenketten: aus $w = w_{i_1} \cdots w_{i_m}$ wird $k_{i_1} \cdots k_{i_m}$.

Eine Verschlüsselungstabelle ist unsicher, wenn es ein Wort w gibt, dessen Kodierung wieder w ist.

Beispiel für eine Verschlüsselungstabelle

Wort	Kodewort
mit	für
der	das
frau	fru
mann	man
an	nanu
hund	katze
dem	den
ein	nein

Aus derhundmitdemmann wird z.B. daskatzefürdenman.

Beispiel für eine Verschlüsselungstabelle

Wort	Kodewort
mit	für
der	das
frau	fru
mann	man
an	nanu
hund	katze
dem	den
ein	nein

Aus derhundmitdemmann wird z.B. daskatzefürdenman.

Ist das Beispiel unsicher?

Beispiel für eine Verschlüsselungstabelle

Wort	Kodewort
mit	für
der	das
frau	fru
mann	man
an	nanu
hund	katze
dem	den
ein	nein

Aus derhundmitdemmann wird z.B. daskatzefürdenman.

Ist das Beispiel unsicher?

Antwort: Ja, da mannein zu mannein wird.

Ein Verschlüsselungsproblems

Ist es entscheidbar, ob eine Verschlüsselungstabelle unsicher ist?

Ein Verschlüsselungsproblems

Ist es entscheidbar, ob eine Verschlüsselungstabelle unsicher ist?

Antwort: Nein, da das Problem zu PCP äquivalent ist.

Das Postsche Korrespondenzproblem

Definition

Gegeben sei ein Alphabet Σ und eine Folge von Wortpaaren

$$K = ((x_1, y_1), \ldots, (x_k, y_k))$$

mit $x_i, y_i \in \Sigma^+$. Das Postsche Korrespondenzproblem (PCP) ist die Frage, ob es für die gegebene Folge K eine nichtleere Folge von Indizes $i_1, \ldots, i_m$ mit $i_j \in \{1, \ldots, k\}$ gibt, sodass

$$x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$$

Aufgabe: PCP-Instanz lösen

Sei $K = \left(\begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix} \right)$ eine PCP-Instanz.

Hat sie eine Lösung?

Aufgabe: PCP-Instanz lösen

Sei $K = \left(\begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix} \right)$ eine PCP-Instanz.

Hat sie eine Lösung?

Antwort:

- Eine Lösung muss mit Stein 2 beginnen (wegen gleichen Anfängen): $\begin{bmatrix} a \\ ab \end{bmatrix}$

Aufgabe: PCP-Instanz lösen

Sei $K = \left(\begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix} \right)$ eine PCP-Instanz.

Hat sie eine Lösung?

Antwort:

- Eine Lösung muss mit Stein 2 beginnen (wegen gleichen Anfängen): $\begin{bmatrix} a \\ ab \end{bmatrix}$
- Danach muss oben ein b kommen. Nur Stein 1 ist möglich: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}$

Aufgabe: PCP-Instanz lösen

Sei $K = \left(\begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix} \right)$ eine PCP-Instanz.

Hat sie eine Lösung?

Antwort:

- Eine Lösung muss mit Stein 2 beginnen (wegen gleichen Anfängen): $\begin{bmatrix} a \\ ab \end{bmatrix}$
- Danach muss oben ein b kommen. Nur Stein 1 ist möglich: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}$
- Oben fehlt ca. Nur Stein 3 ist möglich: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}$

Aufgabe: PCP-Instanz lösen

Sei $K = \left(\begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix} \right)$ eine PCP-Instanz.

Hat sie eine Lösung?

Antwort:

- Eine Lösung muss mit Stein 2 beginnen (wegen gleichen Anfängen): $\begin{bmatrix} a \\ ab \end{bmatrix}$
- Danach muss oben ein b kommen. Nur Stein 1 ist möglich: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}$
- Oben fehlt ca. Nur Stein 3 ist möglich: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}$
- Oben fehlt a. Nur Stein 2 oder 4 ist möglich. Wir nehmen 2: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}$

Aufgabe: PCP-Instanz lösen

Sei $K = \left(\begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix} \right)$ eine PCP-Instanz.

Hat sie eine Lösung?

Antwort:

- Eine Lösung muss mit Stein 2 beginnen (wegen gleichen Anfängen): $\begin{bmatrix} a \\ ab \end{bmatrix}$
- Danach muss oben ein b kommen. Nur Stein 1 ist möglich: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}$
- Oben fehlt ca. Nur Stein 3 ist möglich: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}$
- Oben fehlt a. Nur Stein 2 oder 4 ist möglich. Wir nehmen 2: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}$
- Oben fehlt ab. Nur Stein 2 oder 4 ist möglich. Wir nehmen 4: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix}$

Aufgabe: PCP-Instanz lösen

Sei $K = \left(\begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix} \right)$ eine PCP-Instanz.

Hat sie eine Lösung?

Antwort:

- Eine Lösung muss mit Stein 2 beginnen (wegen gleichen Anfängen): $\begin{bmatrix} a \\ ab \end{bmatrix}$
- Danach muss oben ein b kommen. Nur Stein 1 ist möglich: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}$
- Oben fehlt ca. Nur Stein 3 ist möglich: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}$
- Oben fehlt a. Nur Stein 2 oder 4 ist möglich. Wir nehmen 2: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}$
- Oben fehlt ab. Nur Stein 2 oder 4 ist möglich. Wir nehmen 4: $\begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} b \\ ca \end{bmatrix}, \begin{bmatrix} ca \\ a \end{bmatrix}, \begin{bmatrix} a \\ ab \end{bmatrix}, \begin{bmatrix} abc \\ c \end{bmatrix}$
- Dies führt zur Lösung $2, 1, 3, 2, 4$.

Quiz

Seien $L_1 \subseteq \Sigma_1^*$ und $L_2 \subseteq \Sigma_2^*$ Sprachen, wobei L_1 unentscheidbar ist.
Wie können wir zeigen, dass L_2 unentscheidbar ist?

a) Zeige, dass es eine totale und berechenbare Funktion f gibt mit $w \in L_2$ g.d.w. $f(w) \in L_1$.

b) Zeige, dass es eine totale und berechenbare Funktion f gibt mit $w \in L_1$ g.d.w. $f(w) \in L_2$.

c) Zeige $L_1 \leq L_2$.

d) Zeige $L_2 \leq L_1$.

Quiz

Seien $L_1 \subseteq \Sigma_1^*$ und $L_2 \subseteq \Sigma_2^*$ Sprachen, wobei L_1 unentscheidbar ist.
Wie können wir zeigen, dass L_2 unentscheidbar ist?

a) Zeige, dass es eine totale und berechenbare Funktion f gibt mit $w \in L_2$ g.d.w. $f(w) \in L_1$.

b) Zeige, dass es eine totale und berechenbare Funktion f gibt mit $w \in L_1$ g.d.w. $f(w) \in L_2$.

c) Zeige $L_1 \leq L_2$.

d) Zeige $L_2 \leq L_1$.

Antwort: b) und c).

4. Primitiv und μ-rekursive Funktionen (nur FSK)

Primitiv rekursive Funktionen

Definition

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

(Fortsetzung folgt.)

Primitiv rekursive Funktionen

Definition

- Rekursion: Wenn $g : \mathbb{N}^{k-1} \to \mathbb{N}$ und $h : \mathbb{N}^{k+1} \to \mathbb{N}$ primitiv rekursiv sind, dann ist auch f mit

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

primitiv rekursiv.

Beispiele von primitiv rekursiven Funktionen

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

  ist primitiv rekursiv.

1. Aufgabe: Primitive Rekursion nachweisen

Zeige ausführlich, dass die Summenfunktion mit $sum(0) = 0$ und $sum(n) = sum(n-1) + n$ für $n > 0$ primitiv rekursiv ist.

1. Aufgabe: Primitive Rekursion nachweisen

Zeige ausführlich, dass die Summenfunktion mit $sum(0) = 0$ und $sum(n) = sum(n-1) + n$ für $n > 0$ primitiv rekursiv ist.

Antwort:

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

wobei

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

1. Aufgabe: Primitive Rekursion nachweisen

Zeige ausführlich, dass die Summenfunktion mit $sum(0) = 0$ und $sum(n) = sum(n-1) + n$ für $n > 0$ primitiv rekursiv ist.

Antwort:

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

wobei

- $g() = 0$
- $h(w, x) = ?$

1. Aufgabe: Primitive Rekursion nachweisen

Zeige ausführlich, dass die Summenfunktion mit $sum(0) = 0$ und $sum(n) = sum(n-1) + n$ für $n > 0$ primitiv rekursiv ist.

Antwort:

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

wobei

- $g() = 0$
- $h(w, x) = add(w, add(x, 1))$

1. Aufgabe: Primitive Rekursion nachweisen

Zeige ausführlich, dass die Summenfunktion mit $sum(0) = 0$ und $sum(n) = sum(n-1) + n$ für $n > 0$ primitiv rekursiv ist.

Antwort:

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

wobei

- $g() = 0$
- $h(w, x) = add(w, add(x, 1)) = add(\pi_1^2(w, x), r(w, x))$, wobei
  - $r(w, x) = ?$

1. Aufgabe: Primitive Rekursion nachweisen

Zeige ausführlich, dass die Summenfunktion mit $sum(0) = 0$ und $sum(n) = sum(n-1) + n$ für $n > 0$ primitiv rekursiv ist.

Antwort:

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

wobei

- $g() = 0$
- $h(w, x) = add(w, add(x, 1)) = add(\pi_1^2(w, x), r(w, x))$, wobei
  - $r(w, x) = add(\pi_2^2(w, x), s(w, x))$

1. Aufgabe: Primitive Rekursion nachweisen

Zeige ausführlich, dass die Summenfunktion mit $sum(0) = 0$ und $sum(n) = sum(n-1) + n$ für $n > 0$ primitiv rekursiv ist.

Antwort:

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

wobei

- $g() = 0$
- $h(w, x) = add(w, add(x, 1)) = add(\pi_1^2(w, x), r(w, x))$, wobei
  - $r(w, x) = add(\pi_2^2(w, x), s(w, x))$
  - $s(w, x) = ?$

1. Aufgabe: Primitive Rekursion nachweisen

Zeige ausführlich, dass die Summenfunktion mit $sum(0) = 0$ und $sum(n) = sum(n-1) + n$ für $n > 0$ primitiv rekursiv ist.

Antwort:

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

wobei

- $g() = 0$
- $h(w, x) = add(w, add(x, 1)) = add(\pi_1^2(w, x), r(w, x))$, wobei
  - $r(w, x) = add(\pi_2^2(w, x), s(w, x))$
  - $s(w, x) = 1$

2. Aufgabe: Primitive Rekursion nachweisen

Zeige, dass die Funktion

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

primitiv rekursiv ist.

2. Aufgabe: Primitive Rekursion nachweisen

Zeige, dass die Funktion

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

primitiv rekursiv ist.

Antwort:

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

- Wenn $x \geq y$, dann $sub(y, x) = 0$ und $add(x, 0) = x$.
- Wenn $x < y$, dann $sub(y, x) = y - x$ und $add(x, y - x) = y$.

2. Aufgabe: Primitive Rekursion nachweisen

Zeige, dass die Funktion

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

primitiv rekursiv ist.

Antwort:

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

- Wenn $x \geq y$, dann $sub(y, x) = 0$ und $add(x, 0) = x$.
- Wenn $x < y$, dann $sub(y, x) = y - x$ und $add(x, y - x) = y$.

Da *add*, *sub* und Komposition primitiv rekursiv sind, ist *max* auch primitiv rekursiv.

3. Aufgabe: Primitive Rekursion nachweisen

Zeige, dass die Funktion $\mathit{absdiff}(x, y) = |x - y|$ primitiv rekursiv ist.

3. Aufgabe: Primitive Rekursion nachweisen

Zeige, dass die Funktion $\mathit{absdiff}(x, y) = |x - y|$ primitiv rekursiv ist.

Antwort:

Beachte:

	$\mathit{sub}(x, y)$	$\mathit{sub}(y, x)$	$\mathit{absdiff}(x, y)$
$x \leq y$	0	$y - x$	$y - x$
$x > y$	$x - y$	0	$x - y$

3. Aufgabe: Primitive Rekursion nachweisen

Zeige, dass die Funktion $\mathit{absdiff}(x, y) = |x - y|$ primitiv rekursiv ist.

Antwort:

Beachte:

	$sub(x, y)$	$sub(y, x)$	$\mathit{absdiff}(x, y)$
$x \leq y$	0	$y - x$	$y - x$
$x > y$	$x - y$	0	$x - y$

Daher $\mathit{absdiff}(x, y) = \mathit{add}(sub(x, y), sub(y, x))$.

Da *add*, *sub* und Komposition primitiv rekursiv sind, ist auch *absdiff* primitiv rekursiv.

Primitiv rekursive Prädikate

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

Aufgabe: Primitive Rekursion nachweisen

Zeige, dass das folgende Prädikat primitiv rekursiv ist:

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

Aufgabe: Primitive Rekursion nachweisen

Zeige, dass das folgende Prädikat primitiv rekursiv ist:

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

Antwort:

Beachte: $equal(x, y)$ gilt g.d.w. $|x - y| = 0$. Daher:
$equal(x, y) = eq0?(absdiff(x, y))$, wobei

- $eq0?(x) = \begin{cases} 1 & \text{falls } x = 0 \\ 0 & \text{sonst} \end{cases}$
- D.h. $eq0?(x) = \begin{cases} g() & \text{falls } x = 0 \\ h(eq0?(x-1), x-1) & \text{sonst} \end{cases}$

  wobei
  - $g() = ?$
  - $h(w, x) = ?$

Aufgabe: Primitive Rekursion nachweisen

Zeige, dass das folgende Prädikat primitiv rekursiv ist:

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

Antwort:

Beachte: $equal(x, y)$ gilt g.d.w. $|x - y| = 0$. Daher:
$equal(x, y) = eq0?(absdiff(x, y))$, wobei

- $eq0?(x) = \begin{cases} 1 & \text{falls } x = 0 \\ 0 & \text{sonst} \end{cases}$
- D.h. $eq0?(x) = \begin{cases} g() & \text{falls } x = 0 \\ h(eq0?(x-1), x-1) & \text{sonst} \end{cases}$

  wobei
  - $g() = 1$
  - $h(w, x) = ?$

Aufgabe: Primitive Rekursion nachweisen

Zeige, dass das folgende Prädikat primitiv rekursiv ist:

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

Antwort:

Beachte: $equal(x, y)$ gilt g.d.w. $|x - y| = 0$. Daher:
$equal(x, y) = eq0?(absdiff(x, y))$, wobei

- $eq0?(x) = \begin{cases} 1 & \text{falls } x = 0 \\ 0 & \text{sonst} \end{cases}$
- D.h. $eq0?(x) = \begin{cases} g() & \text{falls } x = 0 \\ h(eq0?(x-1), x-1) & \text{sonst} \end{cases}$

  wobei
  - $g() = 1$
  - $h(w, x) = 0$

μ-Operator

Definition

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

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

Der μ-Operator „sucht“ nach der ersten 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 μh undefiniert.

μ-rekursive Funktionen

Definition

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

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

(Fortsetzung folgt.)

μ-rekursive Funktionen

Definition

- Rekursion: Wenn $g : \mathbb{N}^{k-1} \to \mathbb{N}$ und $h : \mathbb{N}^{k+1} \to \mathbb{N}$ μ-rekursiv sind, dann ist

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

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

Aufgabe: μ-Operator anwenden

Welche Funktion wird durch μh_i mit

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

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

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

berechnet?

Aufgabe: μ-Operator anwenden

Antwort:

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

Aufgabe: μ-Operator anwenden

Antwort:

a) $h_1(x, y) = sub(x, y)$
Da *sub* überall definiert ist, berechnet μh_1 das kleinste x, sodass $sub(x, y) = 0$.

Aufgabe: μ-Operator anwenden

Antwort:

a) $h_1(x, y) = sub(x, y)$
Da sub überall definiert ist, berechnet μh_1 das kleinste x, sodass $sub(x, y) = 0$.
Das gilt schon für $x = 0$, also $(\mu h_1)(y) = 0$.

Aufgabe: μ-Operator anwenden

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

Aufgabe: μ-Operator anwenden

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

Da *sub* überall definiert ist, berechnet μh_2 das kleinste x, sodass $sub(y, x) = 0$

Aufgabe: μ-Operator anwenden

b) $h_2(x, y) = sub(y, x)$
Da *sub* überall definiert ist, berechnet μh_2 das kleinste x, sodass $sub(y, x) = 0$
Dies gilt für $x = y$. Zudem gilt $sub(y, z) > 0$ für alle $z < y$. Daher $(\mu h_2)(y) = y$.

Aufgabe: μ-Operator anwenden

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

Aufgabe: μ-Operator anwenden

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

Wir suchen das kleinste x, sodass $y - 2x \leq 0$.

Aufgabe: μ-Operator anwenden

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

Wir suchen das kleinste x, sodass $y - 2x \leq 0$.

Dies gilt für $x = \lceil y/2 \rceil$. Daher $(\mu h_3)(y) = \lceil y/2 \rceil$.