20.1.2012
Willkommen zur zwölften Übung zur Vorlesung

Grundbegriffe der Informatik

Karlsruhe Institute of Technology

Matthias Janke
email: matthias.janke ät kit.edu

Organisatorisches

- Anmeldung für den Übungsschein nicht vergessen!
- Gestern waren 648 Personen angemeldet
- Anmeldung für die Klausur nicht vergessen!
- Gestern waren 609 Personen angemeldet
- Anmeldung über Studierendenportal: studium.kit.edu
- Online Klausur-Anmeldung möglich für: INFO, INWI, MATH, PHYS

Überblick

Strukturelle Induktion

Turingmaschinen

Strukturelle Induktion - Wörter

- Induktionsanfang: Zeige: X gilt für $w = \epsilon$.
- Induktionsvoraussetzung: Schreibe: X gilt für beliebiges, aber festes $w \in A^*$.
- Induktionsschritt: Schreibe: Sei $x \in A$ beliebig.
  Zeige: Dann gilt X auch für wx.

Strukturelle Induktion - Strukturen

Es gibt atomare Elemente und Operationen, die aus maximal k Elementen ein größeres Element "zusammensetzen".

- Induktionsanfang: Zeige: X gilt für **alle** atomaren Elemente.
- Induktionsvoraussetzung: Schreibe: X gilt für beliebige, aber feste Elemente $e_1, \ldots, e_k$.
- Induktionsschritt: Zeige für **jede** Operation $\circ$ mit $j \leq k$ Argumenten:
  Dann gilt X auch für $\circ(e_1, e_2, \ldots e_j)$.

$Z \subseteq \mathbb{N}_0$
$10, 15 \in Z$
$z_1, z_2 \in Z \Rightarrow z_1 \cdot z_2 \in Z \land z_1 + z_2 \in Z$
Keine anderen Zahlen in Z.

$Z \subseteq \mathbb{N}_0$
$10, 15 \in Z$
$z_1, z_2 \in Z \Rightarrow z_1 \cdot z_2 \in Z \wedge z_1 + z_2 \in Z$
Keine anderen Zahlen in Z.
Zeige: Alle Zahlen $z \in Z$ durch 5 teilbar!

$Z \subseteq \mathbb{N}_0$
$10, 15 \in Z$
$z_1, z_2 \in Z \Rightarrow z_1 \cdot z_2 \in Z \land z_1 + z_2 \in Z$
Keine anderen Zahlen in Z.
IA: 10, 15 durch 5 teilbar $\checkmark$

Strukturelle Induktion - Strukturen

$Z \subseteq \mathbb{N}_0$
$10, 15 \in Z$
$z_1, z_2 \in Z \Rightarrow z_1 \cdot z_2 \in Z \land z_1 + z_2 \in Z$
Keine anderen Zahlen in Z.
IV: Für beliebige, aber feste $z_1, z_2 \in Z$ gilt: z_1, z_2 durch 5 teilbar.

$Z \subseteq \mathbb{N}_0$
$10, 15 \in Z$
$z_1, z_2 \in Z \Rightarrow z_1 \cdot z_2 \in Z \wedge z_1 + z_2 \in Z$
Keine anderen Zahlen in Z.
IS: Dann gilt auch $z_1 + z_2$ durch 5 teilbar und $z_1 \cdot z_2$ durch 5 teilbar (offensichtlich).

Vergleich Endlicher Akzeptor - Turingmaschine

z_0

a a b a

$z_1 = f(z_0, a)$

a a b a

$$z_2 = f(z_1, a)$$

a a b a

Vergleich Endlicher Akzeptor - Turingmaschine

$$z_3 = f(z_2, b)$$

a a b a

$$z_4 = f(z_3, a)$$

a a b a

Vergleich Endlicher Akzeptor - Turingmaschine

	z_0				
□	a	a	b	a	□

$z_1 = f(z_0, a)$

□ a a b a □

$$z_2 = f(z_1, a)$$

□ a a b a □

Vergleich Endlicher Akzeptor - Turingmaschine

$$z_3 = f(z_2, b)$$

$\square$ a a b a $\square$

Vergleich Endlicher Akzeptor - Turingmaschine

$$z_4 = f(z_3, a)$$

□ a a b a □

$$e_{+/-} = f(z_4, \square)$$

$\square$ a a b a $\square$

$A = (Z, z_0, X, f', F)$

$T = (Z \cup \{e_+, e_-\}, z_0, X \cup \{\square\}, f, g, m)$

$$e_{+/-} = f(z_4, \square)$$

$\square$ a a b a $\square$

$A = (Z, z_0, X, f', F)$

$T = (Z \cup \{e_+, e_-\}, z_0, X \cup \{\square\}, f, g, m)$

$f(z, x) = f'(z, x)$

Vergleich Endlicher Akzeptor - Turingmaschine

$$e_{+/-} = f(z_4, \square)$$

$\square$ a a b a $\square$

$A = (Z, z_0, X, f', F)$

$T = (Z \cup \{e_+, e_-\}, z_0, X \cup \{\square\}, f, g, m)$

$f(z, x) = f'(z, x)$

$g(z, x) = \text{egal}$

$$e_{+/-} = f(z_4, \square)$$

$\square$ a a b a $\square$

$A = (Z, z_0, X, f', F)$

$T = (Z \cup \{e_+, e_-\}, z_0, X \cup \{\square\}, f, g, m)$

$f(z, x) = f'(z, x)$

$g(z, x) = \text{egal}$

$m(z, x) = 1$

$$e_{+/-} = f(z_4, \square)$$

$\square$ a a b a $\square$

$A = (Z, z_0, X, f', F)$

$T = (Z \cup \{e_+, e_-\}, z_0, X \cup \{\square\}, f, g, m)$

$$g(z, \square) = \square, m(z, \square) = 0, f(z, \square) = \begin{cases} e_+ & \text{falls } z \in F \\ e_- & \text{falls } z \notin F \end{cases}$$

Berechnung: Zustand über Zeichen:

$$
\begin{array}{cccccc}
 & z_0 & & & & \\
\square & \mathrm{a} & \mathrm{a} & \mathrm{b} & \mathrm{a} & \square \\
 & & z_1 & & & \\
\square & \square & \mathrm{a} & \mathrm{b} & \mathrm{a} & \square \\
 & & & z_2 & & \\
\square & \square & \square & \mathrm{b} & \mathrm{a} & \square \\
 & & & & z_3 & \\
\square & \square & \square & \square & \mathrm{a} & \square \\
 & & & & & z_4 \\
\square & \square & \square & \square & \square & \square \\
 & & & & & e_+ \\
\square & \square & \square & \square & \square & \square
\end{array}
$$

oder Zustand vor Zeichen:

$\square z_0 \text{aaba}\square$

$\square\square z_1 \text{aba}\square$

$\square\square\square z_2 \text{ba}\square$

$\square\square\square\square z_3 \text{a}\square$

$\square\square\square\square\square z_4 \square$

$\square\square\square\square\square e_+ \square$

Gegeben: Problem für Wörter (z.B.: Überprüfe, ob Wort Binärdarstellung einer Primzahl ist)
Gesucht: Turingmaschine, die das Problem löst.
Vorgehensweise: ???

Gegeben: Problem für Wörter (z.B.: Überprüfe, ob Wort Binärdarstellung einer Primzahl ist)
Gesucht: Turingmaschine, die das Problem löst.
Erste Überlegung: Wie würde ich das mit Bleistift und Papier lösen?

Gegeben: Problem für Wörter (z.B.: Überprüfe, ob Wort Binärdarstellung einer Primzahl ist)
Gesucht: Turingmaschine, die das Problem löst.
Erste Überlegung: Wie würde ich das mit Bleistift und Papier lösen?
Überprüfe, ob Wort Binärdarstellung ist!

Gegeben: Problem für Wörter (z.B.: Überprüfe, ob Wort Binärdarstellung einer Primzahl ist)

Gesucht: Turingmaschine, die das Problem löst.

Erste Überlegung: Wie würde ich das mit Bleistift und Papier lösen?

Schreibe nacheinander Wörter $10, 11, 100, 101, 110, 111, \ldots$ hinter ursprüngliches Wort.

Gegeben: Problem für Wörter (z.B.: Überprüfe, ob Wort Binärdarstellung einer Primzahl ist)

Gesucht: Turingmaschine, die das Problem löst.

Erste Überlegung: Wie würde ich das mit Bleistift und Papier lösen?

Schreibe nacheinander Wörter $10, 11, 100, 101, 110, 111, \ldots$ hinter ursprüngliches Wort. Nach einem Trennsymbol $:$.

Gegeben: Problem für Wörter (z.B.: Überprüfe, ob Wort Binärdarstellung einer Primzahl ist)
Gesucht: Turingmaschine, die das Problem löst.
Erste Überlegung: Wie würde ich das mit Bleistift und Papier lösen?
Teile gegebenes Wort durch Wort hinter :

Gegeben: Problem für Wörter (z.B.: Überprüfe, ob Wort Binärdarstellung einer Primzahl ist)

Gesucht: Turingmaschine, die das Problem löst.

Erste Überlegung: Wie würde ich das mit Bleistift und Papier lösen?

Falls Rest 0 und Wort hinter : ungleich w: Nicht prim.

Turing-Maschinen: Design

Gegeben: Problem für Wörter (z.B.: Überprüfe, ob Wort Binärdarstellung einer Primzahl ist)
Gesucht: Turingmaschine, die das Problem löst.
Erste Überlegung: Wie würde ich das mit Bleistift und Papier lösen?
Falls Rest 0 und Wort hinter : gleich w: prim.

Turing-Maschinen: Design

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Binärdarstellung um 1 erhöhen.

Turing-Maschinen: Design

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Binärdarstellung um 1 erhöhen.
Gehe ans rechte Ende des Bandes:
$\forall x \in X \setminus \{\square\} : (f, g, m)(S, x) = (S, x, 1)$
$(f, g, m)(S, \square) = (S_1, \square, -1)$

Turing-Maschinen: Design

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Binärdarstellung um 1 erhöhen.
Addiere 1 zu letzter Ziffer, merke Übertrag:
$(S_1, 0) \mapsto (S_0, 1, -1)$
$(S_1, 1) \mapsto (S_1, 0, -1)$

Turing-Maschinen: Design

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Binärdarstellung um 1 erhöhen.
Addiere Übertrag zu Ziffer, merke Übertrag:
$(S_0, 0) \mapsto (S_0, 0, -1)$
$(S_0, 1) \mapsto (S_0, 1, -1)$

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Binärdarstellung um 1 erhöhen.
Was passiert bei $(S_1, :)$?

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Binärdarstellung um 1 erhöhen.
Was passiert bei $(S_1, :)$?
Entweder Wort vorne dran 1 nach links verschieben,
oder Wort hinter : 1 nach rechts verschieben.

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Binärdarstellung um 1 erhöhen.
Was passiert bei $(S_1, :)$?
Entweder Wort vorne dran 1 nach links verschieben,
oder Wort hinter : 1 nach rechts verschieben.

	Sh	Sh_0	Sh_1	F
0	$(\mathtt{x}, Sh_0, 1)$	$(0, Sh_0, 1)$	$(1, Sh_0, 1)$	-
1	$(\mathtt{x}, Sh_1, 1)$	$(0, Sh_1, 1)$	$(1, Sh_1, 1)$	-
$\square$	-	$(0, F, 0)$	$(1, F, 0)$	-

Jeden Schritt in Turingmaschine übersetzen!

Z.B. Teilen:

$100001 : 11 (w : w')$

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Teilen:
$100001 : 11 (w : w')$
Vergleiche ersten $|w'|$ Ziffern von w mit w'.
Falls Ziffer von w immer $\geq$ Ziffer von w': w' abziehen.

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Teilen:
$100001 : 11 (w : w')$
Vergleiche ersten $|w'|$ Ziffern von w mit w'.
Sonst w' von den ersten $|w'| + 1$ Ziffern von w abziehen.

Turing-Maschinen: Design

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Abziehen:
$100 : 11 (w : w')$
Merke letztes Zeichen; **markiere** letztes Zeichen!

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Abziehen:
$100 : 11 (w : w')$
Merke letztes Zeichen; **markiere** letztes Zeichen!
$(S, 0) \mapsto (S_0, \bar{0}, -1)$
$(S, 1) \mapsto (S_1, \bar{1}, -1)$

Jeden Schritt in Turingmaschine übersetzen!

Z.B. Abziehen:

$100 : 11 (w : w')$

Gehe zu Wort vor dem :

$\forall x \in X \setminus \{:\} \forall i \in \mathbb{G}_2 : (S_i, x) \mapsto (S_i, x, -1)$

$(S_i, :) \mapsto (Z_i, :, -1)$

Jeden Schritt in Turingmaschine übersetzen!

Z.B. Abziehen:

$100 : 11 (w : w')$

Gehe zu erster unmarkierter Ziffer vor dem :

$(Z_i, \bar{0}) \mapsto (Z_i, \bar{0}, -1)$

$(Z_i, \bar{1}) \mapsto (Z_i, \bar{1}, -1)$

Jeden Schritt in Turingmaschine übersetzen!

Z.B. Abziehen:

$100 : 11 (w : w')$

Ziehe Ziffern von einander ab; merke Übertrag:

$(Z_0, i) \mapsto (U_0, \bar{i}, 1)$

$(Z_1, 1) \mapsto (U_0, \bar{0}, 1)$ $(Z_1, 0) \mapsto (U_1, \bar{1}, 1)$

Jeden Schritt in Turingmaschine übersetzen!

Z.B. Abziehen:

$100 : 11 (w : w')$

Gehe zu letzter nicht markierter Ziffer nach :

Jeden Schritt in Turingmaschine übersetzen!

Z.B. Abziehen:

$100 : 11(w : w')$

Addiere Übertrag zu gelesener Ziffer:

$(U'_i, j) \mapsto (S_{i+j}, \bar{j}, -1)$

Turing-Maschinen: Design

Jeden Schritt in Turingmaschine übersetzen!

Z.B. Abziehen:

$100 : 11 (w : w')$

Wenn Übertrag 1 und keine unmarkierte Ziffer hinter :

Jeden Schritt in Turingmaschine übersetzen!
Z.B. Abziehen:
$100 : 11 (w : w')$
Wenn Übertrag 1 und keine unmarkierte Ziffer hinter :
$\rightarrow$ Fahre zur letzten unmarkierten Ziffer des ersten Wortes, ziehe 1 ab.

Merke:

- Gelesene Zeichen in Index speichern hilft, zu wissen, was man eigentlich tun will!
- Gelesene Zeichen als gelesen markieren entspricht "sich merken, wo man eigentlich gerade war".

1. Finde Zustände, bei denen Turingmaschine einfach zum rechten/linken Ende des Wortes fährt.
2. Überprüfe Zustandsnamen auf Hinweise, was gespeichert wird.
3. Führe Berechnung an Beispiel durch. (Hinweis: sofern nicht alle Zwischenschritte gefordert sind, kann man mit 1. abkürzen.)
4. Formuliere These, was Turingmaschine in einzelnen Zuständen macht.
5. Herausfinden, was die Turingmaschine an sich macht.

Eingabealphabet $\{a\}$, Bandalphabet $\{a, b, 0, 1, \square\}$,
Anfangszustand z_0

	z_0	z_1	r	w
a	$(z_1, b, 1)$	$(z_0, a, 1)$	$(w, a, -1)$	$(w, a, -1)$
b	$(z_0, b, 1)$	$(z_1, b, 1)$	$(r, b, -1)$	$(w, b, -1)$
0	$(z_0, 0, 1)$	$(z_1, 0, 1)$	$(r, 0, -1)$	$(w, 0, -1)$
1	$(z_0, 1, 1)$	$(z_1, 1, 1)$	$(r, 1, -1)$	$(w, 1, -1)$
$\square$	$(r, 0, -1)$	$(r, 1, -1)$	-	$(z_0, \square, 1)$

Feststellungen:

1. w läuft nach links durch.

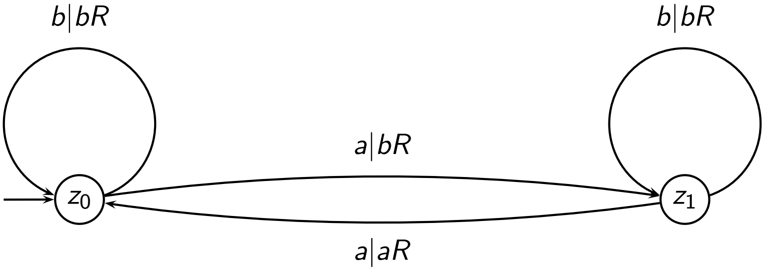
Feststellungen:

1. w läuft nach links durch.
2. r läuft nach links durch, bis es auf a trifft; wird dann zu w

Feststellungen:

1. w läuft nach links durch.
2. r läuft nach links durch, bis es auf a trifft; wird dann zu w
3. r überprüft, ob noch a in Wort vorhanden; falls nicht, Ende.

z_0 und z_1:



Feststellungen:

1. Das i bei z_i ist die Anzahl der gelesenen a mod 2.

Feststellungen:

1. Das i bei z_i ist die Anzahl der gelesenen $a \mod 2$.
2. Wenn keine a mehr kommen, läuft z_i nach rechts.

Feststellungen:

1. Das i bei z_i ist die Anzahl der gelesenen $a \mod 2$.
2. Wenn keine a mehr kommen, läuft z_i nach rechts.
3. z_i schreibt i an Ende des Wortes.

Feststellungen:

Anfang a^n auf Band.

1. Anzahl der a nach Rückkehr des Kopfes $\lfloor \frac{n}{2} \rfloor$.

Feststellungen:
Anfang a^n auf Band.
1. Anzahl der a nach Rückkehr des Kopfes $\lfloor \frac{n}{2} \rfloor$.
2. Ans Ende wird geschrieben $n \mod 2, \lfloor \frac{n}{2} \rfloor \mod 2, \lfloor \frac{n}{4} \rfloor \mod 2 \ldots$

Feststellungen:
Anfang a^n auf Band.
1. Anzahl der a nach Rückkehr des Kopfes $\lfloor \frac{n}{2} \rfloor$.
2. Ans Ende wird geschrieben $n \mod 2, \lfloor \frac{n}{2} \rfloor \mod 2, \lfloor \frac{n}{4} \rfloor \mod 2 \ldots$
3. Vorstellung von n als Binärzahl: n wird binär rückwärts ans Ende geschrieben.

Feststellungen:

Anfang a^n auf Band.

1. Anzahl der a nach Rückkehr des Kopfes $\lfloor \frac{n}{2} \rfloor$.
2. Ans Ende wird geschrieben $n \mod 2, \lfloor \frac{n}{2} \rfloor \mod 2, \lfloor \frac{n}{4} \rfloor \mod 2 \ldots$
3. Vorstellung von n als Binärzahl: n wird binär rückwärts ans Ende geschrieben.
4. Am Ende auf Band: $b^n R(Repr_2(n))$.

Überlegen Sie sich, wie sie ein Problem mit einer Turingmaschine lösen würden!
Entwerfen Sie eine Turingmaschine!
Tauschen Sie Turingmaschinen untereinander und finden Sie heraus, was die Turingmaschinen Ihrer Kommilitonen machen!

Das wars für heute...

Themen für das zwölfte Übungsblatt:

- Strukturelle Induktion
- Turingmaschinen

Schönes Wochenende!