Einführung in die Theoretische Informatik

Christian Dalvit   Manuel Eberl
Samuel Frontull   **Cezary Kaliszyk**   Daniel Ranalter

Wintersemester 2022/23

universität innsbruck

Frohes Neues!

Wintersemester 2022/23

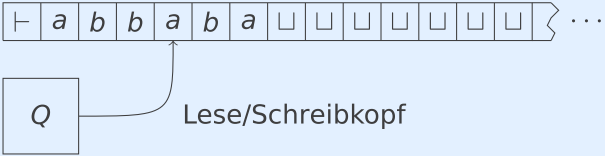
universität innsbruck

Zusammenfassung

Wintersemester 2022/23

Definition (informell)

deterministische, einbändige Turingmaschine (TM):



- Eine TM verwendet ein einseitig unendliches Band als Speicher
- Zu Beginn der Berechnung steht die Eingabe auf dem Band
- Der Speicher wird mit einem Lese/Schreibkopf gelesen oder beschrieben
- Das Verhalten der TM wird durch die endliche Kontrolle Q kontrolliert

Unentscheidbarkeit

Satz

Es kann niemals ein Testprogramm für „hello, world"-Programme geben

Definition (informell)

ein Problem, das nicht algorithmisch lösbar ist, heißt unentscheidbar

Satz

die folgenden Probleme sind unentscheidbar:

1. *das Halteproblem*
2. *das Postsche Korrespondenzproblem*
3. *ist eine beliebige kontextfreie Grammatik eindeutig*
4. *. . .*

Einführung in die Logik

Syntax & Semantik der Aussagenlogik, Formales Beweisen, Konjunktive und Disjunktive Normalformen

Einführung in die Algebra

Algebraische Strukturen, Boolesche Algebra, Universelle Algebra

Einführung in die Theorie der Formalen Sprachen

Grammatiken und Formale Sprachen, Chomsky-Hierarchie, Reguläre Sprachen, Kontextfreie Sprachen, Anwendungen von formalen Sprachen

Einführung in die Berechenbarkeitstheorie und Komplexitätstheorie

Algorithmisch unlösbare Probleme, Turing Maschinen, Registermaschinen, Komplexitätstheorie

Einführung in die Programmverifikation

Prinzipien der Analyse von Programmen, Verifikation nach Hoare

Einführung in die Logik

Syntax & Semantik der Aussagenlogik, Formales Beweisen, Konjunktive und Disjunktive Normalformen

Einführung in die Algebra

Algebraische Strukturen, Boolesche Algebra, Universelle Algebra

Einführung in die Theorie der Formalen Sprachen

Grammatiken und Formale Sprachen, Chomsky-Hierarchie, Reguläre Sprachen, Kontextfreie Sprachen, Anwendungen von formalen Sprachen

Einführung in die Berechenbarkeitstheorie und Komplexitätstheorie

Algorithmisch unlösbare Probleme, Turing Maschinen, Registermaschinen, Komplexitätstheorie

Einführung in die Programmverifikation

Prinzipien der Analyse von Programmen, Verifikation nach Hoare

Beispiel (Wiederholung)

sei $M = (\{s, t, r, q_1, q_2, q_3\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1, \mathrm{X}, \mathrm{Y}\}, \vdash, \sqcup, \delta, s, t, r)$ mit δ wie folgt

Beispiel (Wiederholung)

sei $M = (\{s,t,r,q_1,q_2,q_3\},\{0,1\},\{\vdash,\sqcup,0,1,\mathsf{X},\mathsf{Y}\},\vdash,\sqcup,\delta,s,t,r)$ mit δ wie folgt

$p \in Q$	$a \in \Gamma$	$\delta(p,a)$	$p \in Q$	$a \in \Gamma$	$\delta(p,a)$
s	$\vdash$	$(s,\vdash,\mathsf{R})$	q_2	$\vdash$	$(r,\vdash,\mathsf{R})$
s	$\sqcup$	$(r,\sqcup,\mathsf{R})$	q_2	$\sqcup$	$(r,\sqcup,\mathsf{R})$
s	0	$(q_1,\mathsf{X},\mathsf{R})$	q_2	0	$(q_2,0,\mathsf{L})$
s	1	$(r,1,\mathsf{R})$	q_2	1	$(r,1,\mathsf{R})$
s	X	$(r,\mathsf{X},\mathsf{R})$	q_2	X	$(s,\mathsf{X},\mathsf{R})$
s	Y	$(q_3,\mathsf{Y},\mathsf{R})$	q_2	Y	$(q_2,\mathsf{Y},\mathsf{L})$
q_1	$\vdash$	$(r,\vdash,\mathsf{R})$	q_3	$\vdash$	$(r,\vdash,\mathsf{R})$
q_1	$\sqcup$	$(r,\vdash,\mathsf{R})$	q_3	$\sqcup$	$(t,\sqcup,\mathsf{R})$
q_1	0	$(q_1,0,\mathsf{R})$	q_3	0	$(r,0,\mathsf{R})$
q_1	1	$(q_2,\mathsf{Y},\mathsf{L})$	q_3	1	$(r,1,\mathsf{R})$
q_1	X	$(r,\mathsf{X},\mathsf{R})$	q_3	X	$(r,\mathsf{X},\mathsf{R})$
q_1	Y	$(q_1,\mathsf{Y},\mathsf{R})$	q_3	Y	$(q_3,\mathsf{Y},\mathsf{R})$
t	$*$	$(t,*,\mathsf{R})$	r	$*$	$(r,*,\mathsf{R})$

Beispiel (Wiederholung)

sei $M = (\{s, t, r, q_1, q_2, q_3\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1, \mathsf{X}, \mathsf{Y}\}, \vdash, \sqcup, \delta, s, t, r)$ mit δ wie folgt

$p \in Q$	$a \in \Gamma$	$\delta(p,a)$	$p \in Q$	$a \in \Gamma$	$\delta(p,a)$
s	$\vdash$	$(s, \vdash, \mathsf{R})$			
s	0	$(q_1, \mathsf{X}, \mathsf{R})$	q_2	0	$(q_2, 0, \mathsf{L})$
			q_2	X	$(s, \mathsf{X}, \mathsf{R})$
s	Y	$(q_3, \mathsf{Y}, \mathsf{R})$	q_2	Y	$(q_2, \mathsf{Y}, \mathsf{L})$
			q_3	$\sqcup$	$(t, \sqcup, \mathsf{R})$
q_1	0	$(q_1, 0, \mathsf{R})$			
q_1	1	$(q_2, \mathsf{Y}, \mathsf{L})$			
q_1	Y	$(q_1, \mathsf{Y}, \mathsf{R})$	q_3	Y	$(q_3, \mathsf{Y}, \mathsf{R})$

Beispiel

Wir betrachten die Schrittfunktion für eine akzeptierende Berechnung von M bei Eingabe 0011:

Beispiel

Wir betrachten die Schrittfunktion für eine akzeptierende Berechnung von M bei Eingabe 0011:

$$(s, \vdash 0011\sqcup^\infty, 0) \xrightarrow[M]{1} (s, \vdash 0011\sqcup^\infty, 1)$$

Beispiel

Wir betrachten die Schrittfunktion für eine akzeptierende Berechnung von M bei Eingabe 0011:

$$(s, \vdash 0011\sqcup^\infty, 0) \xrightarrow[M]{1} (s, \vdash 0011\sqcup^\infty, 1) \xrightarrow[M]{1} (q_1, \vdash X011\sqcup^\infty, 2)$$

Beispiel

Wir betrachten die Schrittfunktion für eine akzeptierende Berechnung von M bei Eingabe 0011:

$$(s, \vdash 0011\sqcup^{\infty}, 0) \xrightarrow[M]{1} (s, \vdash 0011\sqcup^{\infty}, 1) \xrightarrow[M]{1} (q_1, \vdash \mathrm{X}011\sqcup^{\infty}, 2)$$
$$\xrightarrow[M]{1} (q_1, \vdash \mathrm{X}011\sqcup^{\infty}, 3)$$

Beispiel

Wir betrachten die Schrittfunktion für eine akzeptierende Berechnung von M bei Eingabe 0011:

$$
\begin{aligned}
(s, \vdash 0011\sqcup^\infty, 0) \xrightarrow[M]{1} (s, \vdash 0011\sqcup^\infty, 1) \xrightarrow[M]{1} (q_1, \vdash \mathrm{X}011\sqcup^\infty, 2) \\
\xrightarrow[M]{1} (q_1, \vdash \mathrm{X}011\sqcup^\infty, 3) \xrightarrow[M]{1} (q_2, \vdash \mathrm{X}0\mathrm{Y}1\sqcup^\infty, 2)
\end{aligned}
$$

Beispiel

Wir betrachten die Schrittfunktion für eine akzeptierende Berechnung von M bei Eingabe 0011:

$$\begin{aligned}(s, \vdash 0011\sqcup^\infty, 0) \xrightarrow[M]{1} (s, \vdash 0011\sqcup^\infty, 1) &\xrightarrow[M]{1} (q_1, \vdash X011\sqcup^\infty, 2)\\ &\xrightarrow[M]{1} (q_1, \vdash X011\sqcup^\infty, 3) \xrightarrow[M]{1} (q_2, \vdash X0Y1\sqcup^\infty, 2)\\ &\xrightarrow[M]{1} (q_2, \vdash X0Y1\sqcup^\infty, 1)\end{aligned}$$

Beispiel

Wir betrachten die Schrittfunktion für eine akzeptierende Berechnung von M bei Eingabe 0011:

$$
\begin{aligned}
(s, \vdash 0011\sqcup^{\infty}, 0) \xrightarrow[M]{1} (s, \vdash 0011\sqcup^{\infty}, 1) &\xrightarrow[M]{1} (q_1, \vdash \mathrm{X}011\sqcup^{\infty}, 2) \\
&\xrightarrow[M]{1} (q_1, \vdash \mathrm{X}011\sqcup^{\infty}, 3) \xrightarrow[M]{1} (q_2, \vdash \mathrm{X0Y1}\sqcup^{\infty}, 2) \\
&\xrightarrow[M]{1} (q_2, \vdash \mathrm{X0Y1}\sqcup^{\infty}, 1) \xrightarrow[M]{1} (s, \vdash \mathrm{X0Y1}\sqcup^{\infty}, 2)
\end{aligned}
$$

Beispiel

Wir betrachten die Schrittfunktion für eine akzeptierende Berechnung von M bei Eingabe 0011:

$$
\begin{aligned}
(s, \vdash 0011\sqcup^\infty, 0) &\xrightarrow[M]{1} (s, \vdash 0011\sqcup^\infty, 1) \xrightarrow[M]{1} (q_1, \vdash X011\sqcup^\infty, 2) \\
&\xrightarrow[M]{1} (q_1, \vdash X011\sqcup^\infty, 3) \xrightarrow[M]{1} (q_2, \vdash X0Y1\sqcup^\infty, 2) \\
&\xrightarrow[M]{1} (q_2, \vdash X0Y1\sqcup^\infty, 1) \xrightarrow[M]{1} (s, \vdash X0Y1\sqcup^\infty, 2) \\
&\xrightarrow[M]{1} (q_1, \vdash XXY1\sqcup^\infty, 3) \xrightarrow[M]{1} (q_1, \vdash XXY1\sqcup^\infty, 4) \\
&\xrightarrow[M]{1} (q_2, \vdash XXYY\sqcup^\infty, 3) \xrightarrow[M]{1} (q_2, \vdash XXYY\sqcup^\infty, 2) \\
&\xrightarrow[M]{1} (s, \vdash XXYY\sqcup^\infty, 3) \xrightarrow[M]{1} (q_3, \vdash XXYY\sqcup^\infty, 4) \\
&\xrightarrow[M]{1} (q_3, \vdash XXYY\sqcup^\infty, 5) \xrightarrow[M]{1} (t, \vdash XXYY\sqcup\sqcup^\infty, 6)
\end{aligned}
$$

Beispiel

Wir betrachten die Schrittfunktion für eine akzeptierende Berechnung von M bei Eingabe 0011:

$$
\begin{aligned}
(s, \vdash 0011\sqcup^\infty, 0) &\xrightarrow[M]{1} (s, \vdash 0011\sqcup^\infty, 1) \xrightarrow[M]{1} (q_1, \vdash X011\sqcup^\infty, 2)\\
&\xrightarrow[M]{1} (q_1, \vdash X011\sqcup^\infty, 3) \xrightarrow[M]{1} (q_2, \vdash X0Y1\sqcup^\infty, 2)\\
&\xrightarrow[M]{1} (q_2, \vdash X0Y1\sqcup^\infty, 1) \xrightarrow[M]{1} (s, \vdash X0Y1\sqcup^\infty, 2)\\
&\xrightarrow[M]{1} (q_1, \vdash XXY1\sqcup^\infty, 3) \xrightarrow[M]{1} (q_1, \vdash XXY1\sqcup^\infty, 4)\\
&\xrightarrow[M]{1} (q_2, \vdash XXYY\sqcup^\infty, 3) \xrightarrow[M]{1} (q_2, \vdash XXYY\sqcup^\infty, 2)\\
&\xrightarrow[M]{1} (s, \vdash XXYY\sqcup^\infty, 3) \xrightarrow[M]{1} (q_3, \vdash XXYY\sqcup^\infty, 4)\\
&\xrightarrow[M]{1} (q_3, \vdash XXYY\sqcup^\infty, 5) \xrightarrow[M]{1} (t, \vdash XXYY \sqcup \sqcup^\infty, 6)
\end{aligned}
$$

Definition

eine TM M

- akzeptiert $x \in \Sigma^*$, wenn $\exists\, y, n$:

$$(s, \vdash x \sqcup^\infty, 0) \xrightarrow[M]{*} (t, y, n)$$

- verwirft $x \in \Sigma^*$, wenn $\exists\, y, n$:

$$(s, \vdash x \sqcup^\infty, 0) \xrightarrow[M]{*} (r, y, n)$$

Definition

eine TM M

- akzeptiert $x \in \Sigma^*$, wenn $\exists\, y$, n:
$$(s, \vdash x \sqcup^\infty, 0) \xrightarrow[M]{*} (t, y, n)$$
- verwirft $x \in \Sigma^*$, wenn $\exists\, y$, n:
$$(s, \vdash x \sqcup^\infty, 0) \xrightarrow[M]{*} (r, y, n)$$
- hält bei Eingabe x, wenn x akzeptiert oder verworfen
- hält nicht bei Eingabe x, wenn x weder akzeptiert, noch verworfen

Definition

eine TM M

- akzeptiert $x \in \Sigma^*$, wenn $\exists\, y, n$:
$$(s, \vdash x \sqcup^\infty, 0) \xrightarrow[M]{*} (t, y, n)$$
- verwirft $x \in \Sigma^*$, wenn $\exists\, y, n$:
$$(s, \vdash x \sqcup^\infty, 0) \xrightarrow[M]{*} (r, y, n)$$
- hält bei Eingabe x, wenn x akzeptiert oder verworfen
- hält nicht bei Eingabe x, wenn x weder akzeptiert, noch verworfen
- ist total, wenn M auf allen Eingaben hält

Definition

eine TM M

- akzeptiert $x \in \Sigma^*$, wenn $\exists\, y, n$:
$$(s, \vdash x \sqcup^\infty, 0) \xrightarrow[M]{*} (t, y, n)$$
- verwirft $x \in \Sigma^*$, wenn $\exists\, y, n$:
$$(s, \vdash x \sqcup^\infty, 0) \xrightarrow[M]{*} (r, y, n)$$
- hält bei Eingabe x, wenn x akzeptiert oder verworfen
- hält nicht bei Eingabe x, wenn x weder akzeptiert, noch verworfen
- ist total, wenn M auf allen Eingaben hält

Definition

die Sprache einer TM M ist wie folgt definiert:

$$\mathrm{L}(M) := \{x \in \Sigma^* \mid M \text{ akzeptiert } x\}$$

Satz

Sei L eine Sprache, die von einer TM akzeptiert wird. Dann ist L rekursiv aufzählbar.

Satz

Sei L eine Sprache, die von einer TM akzeptiert wird. Dann ist L rekursiv aufzählbar.

Folgerung

Sei L eine Sprache, die von einer TM akzeptiert wird, dann existiert eine Grammatik G, sodass $L = \mathrm{L}(G)$

Satz

Sei L eine Sprache, die von einer TM akzeptiert wird. Dann ist L rekursiv aufzählbar.

Folgerung

Sei L eine Sprache, die von einer TM akzeptiert wird, dann existiert eine Grammatik G, sodass $L = \mathsf{L}(G)$

Definition (Berechenbarkeit mit einer TM)

- Sei $M = (Q, \{\sqcap, \square\}, \{\vdash, \sqcup, \sqcap, \square\}, \vdash, \sqcup, \delta, s, t, r)$

Satz

Sei L eine Sprache, die von einer TM akzeptiert wird. Dann ist L rekursiv aufzählbar.

Folgerung

Sei L eine Sprache, die von einer TM akzeptiert wird, dann existiert eine Grammatik G, sodass $L = \mathrm{L}(G)$

Definition (Berechenbarkeit mit einer TM)

- Sei $M = (Q, \{\sqcap, \square\}, \{\vdash, \sqcup, \sqcap, \square\}, \vdash, \sqcup, \delta, s, t, r)$
- Eine partielle Funktion $f \colon \mathbb{N}^k \to \mathbb{N}$ heißt M-berechenbar, wenn:

$$f(n_1, \dots, n_k) = m \quad \text{gdw.} \quad (s, \vdash \sqcap^{n_1} \square \cdots \square \sqcap^{n_k} \sqcup^{\infty}, 0) \xrightarrow[M]{*} (t, \vdash \sqcap^{m} \sqcup^{\infty}, n)$$

Satz

Sei L eine Sprache, die von einer TM akzeptiert wird. Dann ist L rekursiv aufzählbar.

Folgerung

Sei L eine Sprache, die von einer TM akzeptiert wird, dann existiert eine Grammatik G, sodass $L = \mathrm{L}(G)$

Definition (Berechenbarkeit mit einer TM)

- Sei $M = (Q, \{\sqcap, \square\}, \{\vdash, \sqcup, \sqcap, \square\}, \vdash, \sqcup, \delta, s, t, r)$
- Eine partielle Funktion $f\colon \mathbb{N}^k \to \mathbb{N}$ heißt M-berechenbar, wenn:

$$f(n_1, \dots, n_k) = m \quad \text{gdw.} \quad (s, \vdash \sqcap^{n_1} \square \cdots \square \sqcap^{n_k} \sqcup^\infty, 0) \xrightarrow[M]{*} (t, \vdash \sqcap^m \sqcup^\infty, n)$$

- Eine partielle Funktion $f\colon \mathbb{N}^k \to \mathbb{N}$ heißt berechenbar mit einer TM, wenn eine TM M über dem Alphabet $\{\sqcap, \square\}$ existiert, sodass f M-berechenbar

Turing Maschinen Berechenbarkeit

Beispiel

Wir beschreiben informell eine TM M, die zwei Zahlen x_1 und x_2 als Eingabe bekommt; M soll $x_1 - x_2$ berechnen und diese Differenz verdoppeln

Turing Maschinen Berechenbarkeit

Beispiel

Wir beschreiben informell eine TM M, die zwei Zahlen x_1 und x_2 als Eingabe bekommt; M soll $x_1 - x_2$ berechnen und diese Differenz verdoppeln

Algorithmus

- wir kodieren die Eingabe x_1, x_2 durch Wörter über dem Alphabet $\{\sqcap\}$ der Länge x_1 bzw. x_2
- die beiden Eingaben werden durch Trennsymbol $\square$ getrennt
- Subtraktion funktioniert, indem wir sukzessive für alle $\sqcap$ rechts von $\square$ ein $\sqcap$ links mit $\sqcup$ überschreiben
- Verdopplung funktioniert indem wir für jedes übrige $\sqcap$, $\sqcap\sqcap$ schreiben

Registermaschinen

Definition

Eine Registermaschine (RM) R ist ein Paar $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathsf{P})$ sodass

Registermaschinen

Definition

Eine Registermaschine (RM) R ist ein Paar $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathsf{P})$ sodass

1. $(x_i)_{1 \leqslant i \leqslant n}$ eine Sequenz von n Registern x_i, die natürliche Zahlen beinhalten

Registermaschinen

Definition

Eine Registermaschine (RM) R ist ein Paar $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathsf{P})$ sodass

1. $(x_i)_{1 \leqslant i \leqslant n}$ eine Sequenz von n Registern x_i, die natürliche Zahlen beinhalten
2. P ein Programm

Registermaschinen

Definition

Eine Registermaschine (RM) R ist ein Paar $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathrm{P})$ sodass

1. $(x_i)_{1 \leqslant i \leqslant n}$ eine Sequenz von n Registern x_i, die natürliche Zahlen beinhalten
2. P ein Programm

Programme sind endliche Folgen von Befehlen und sind induktiv definiert:

Registermaschinen

Definition

Eine Registermaschine (RM) R ist ein Paar $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathsf{P})$ sodass

1. $(x_i)_{1 \leqslant i \leqslant n}$ eine Sequenz von n Registern x_i, die natürliche Zahlen beinhalten
2. P ein Programm

Programme sind endliche Folgen von Befehlen und sind induktiv definiert:

1. Für jedes Register x_i sind die folgenden Instruktionen sowohl Befehle wie Programme: $x_i := x_i + 1$ und $x_i := x_i - 1$

Registermaschinen

Definition

Eine Registermaschine (RM) R ist ein Paar $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathrm{P})$ sodass

1. $(x_i)_{1 \leqslant i \leqslant n}$ eine Sequenz von n Registern x_i, die natürliche Zahlen beinhalten
2. P ein Programm

Programme sind endliche Folgen von Befehlen und sind induktiv definiert:

1. Für jedes Register x_i sind die folgenden Instruktionen sowohl Befehle wie Programme: $x_i := x_i + 1$ und $x_i := x_i - 1$
2. wenn P_1, P_2 Programme sind, dann ist $P_1; P_2$ ein Programm

Registermaschinen

Definition

Eine Registermaschine (RM) R ist ein Paar $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathrm{P})$ sodass

1. $(x_i)_{1 \leqslant i \leqslant n}$ eine Sequenz von n Registern x_i, die natürliche Zahlen beinhalten
2. P ein Programm

Programme sind endliche Folgen von Befehlen und sind induktiv definiert:

1. Für jedes Register x_i sind die folgenden Instruktionen sowohl Befehle wie Programme: $x_i := x_i + 1$ und $x_i := x_i - 1$
2. wenn P_1, P_2 Programme sind, dann ist $P_1; P_2$ ein Programm
3. wenn P_1 ein Programm und x_i ein Register, dann ist

$$\text{while } x_i \neq 0 \text{ do } P_1 \text{ end}$$

   sowohl ein Befehl als auch ein Programm

Definition (Semantik von Registermaschinen)

1. Zu Beginn der Berechnung steht die Eingabe (als natürliche Zahlen) in den Registern

Definition (Semantik von Registermaschinen)

1. Zu Beginn der Berechnung steht die Eingabe (als natürliche Zahlen) in den Registern
2. Die Befehle
   - $x_i := x_i + 1$
   - $x_i := x_i - 1$

   bedeuten, dass der Inhalt des Register x_i entweder um 1 erhöht oder vermindert wird

Definition (Semantik von Registermaschinen)

1. Zu Beginn der Berechnung steht die Eingabe (als natürliche Zahlen) in den Registern
2. Die Befehle
   - $x_i := x_i + 1$
   - $x_i := x_i - 1$

   bedeuten, dass der Inhalt des Register x_i entweder um 1 erhöht oder vermindert wird
3. $P_1; P_2$ bedeutet, dass zunächst das Programm P_1 und dann das Programm P_2 ausgeführt wird

Definition (Semantik von Registermaschinen)

1. Zu Beginn der Berechnung steht die Eingabe (als natürliche Zahlen) in den Registern
2. Die Befehle
   - $x_i := x_i + 1$
   - $x_i := x_i - 1$

   bedeuten, dass der Inhalt des Register x_i entweder um 1 erhöht oder vermindert wird
3. $P_1; P_2$ bedeutet, dass zunächst das Programm P_1 und dann das Programm P_2 ausgeführt wird
4. Der Befehl (und das Programm)

$$\text{while } x_i \neq 0 \text{ do } P_1 \text{ end}$$

   bedeutet, der Schleifenrumpf P_1 wird ausgeführt, bis die Bedingung $x_i \neq 0$ falsch ist

Beispiel

Sei $R = ((x_i)_{1 \leqslant i \leqslant 5}, \mathrm{P})$ eine RM mit dem folgenden Programm:

Beispiel

Sei $R = ((x_i)_{1 \leqslant i \leqslant 5}, \mathrm{P})$ eine RM mit dem folgenden Programm:

$$
\begin{array}{l}
x_3 := 0; \\
\text{while } x_1 \neq 0 \text{ do} \\
\quad x_1 := x_1 - 1; \\
\quad x_4 := x_2; \\
\quad \text{while } x_2 \neq 0 \text{ do} \\
\quad\quad x_2 := x_2 - 1; \\
\quad\quad x_3 := x_3 + 1 \\
\quad \text{end}; \\
\quad x_2 := x_4 \\
\text{end}
\end{array}
$$

Beispiel

Sei $R = ((x_i)_{1 \leqslant i \leqslant 5}, \mathrm{P})$ eine RM mit dem folgenden Programm:

Multiplikation

$x_3 := 0$;
while $x_1 \neq 0$ do
  $x_1 := x_1 - 1$;
  $x_4 := x_2$;
  while $x_2 \neq 0$ do
    $x_2 := x_2 - 1$;
    $x_3 := x_3 + 1$
  end;
  $x_2 := x_4$
end

Beispiel

Sei $R = ((x_i)_{1 \leqslant i \leqslant 5}, \mathrm{P})$ eine RM mit dem folgenden Programm:

Zuweisung $x_i := x_j$

$\text{while } x_i \neq 0 \text{ do}$
$\quad x_i := x_i - 1$
end;
$\text{while } x_k \neq 0 \text{ do}$
$\quad x_k := x_k - 1$
end

$\text{while } x_j \neq 0 \text{ do}$
$\quad x_i := x_i + 1;$
$\quad x_j := x_j - 1;$
$\quad x_k := x_k + 1$
end;
$\text{while } x_k \neq 0 \text{ do}$
$\quad x_j := x_j + 1;$
$\quad x_k := x_k - 1$
end

Multiplikation

$x_3 := 0;$
$\text{while } x_1 \neq 0 \text{ do}$
$\quad x_1 := x_1 - 1;$
$\quad x_4 := x_2;$
$\quad \text{while } x_2 \neq 0 \text{ do}$
$\quad\quad x_2 := x_2 - 1;$
$\quad\quad x_3 := x_3 + 1$
$\quad \text{end;}$
$\quad x_2 := x_4$
end

Beispiel

Sei $R = ((x_i)_{1 \leqslant i \leqslant 5}, \mathrm{P})$ eine RM mit dem folgenden Programm:

Zuweisung $x_i := x_j$

$$
\begin{array}{l}
\text{while } x_i \neq 0 \text{ do} \\
\quad x_i := x_i - 1 \\
\text{end;} \\
\text{while } x_k \neq 0 \text{ do} \\
\quad x_k := x_k - 1 \\
\text{end}
\end{array}
$$

$$
\begin{array}{l}
\text{while } x_j \neq 0 \text{ do} \\
\quad x_i := x_i + 1; \\
\quad x_j := x_j - 1; \\
\quad x_k := x_k + 1 \\
\text{end;} \\
\text{while } x_k \neq 0 \text{ do} \\
\quad x_j := x_j + 1; \\
\quad x_k := x_k - 1 \\
\text{end}
\end{array}
$$

Multiplikation

$$
\begin{array}{l}
x_3 := 0; \\
\text{while } x_1 \neq 0 \text{ do} \\
\quad x_1 := x_1 - 1; \\
\quad x_4 := x_2; \\
\quad \text{while } x_2 \neq 0 \text{ do} \\
\quad\quad x_2 := x_2 - 1; \\
\quad\quad x_3 := x_3 + 1 \\
\quad \text{end;} \\
\quad x_2 := x_4 \\
\text{end}
\end{array}
$$

Bei Eingabe $(m, n, 0, 0, 0)$ berechnet R $(0, n, m \times n, n, 0)$

Berechenbarkeit mit einer RM

Definition

- sei $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathrm{P})$ eine RM

Berechenbarkeit mit einer RM

Definition

- sei $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathsf{P})$ eine RM
- eine partielle Funktion $f \colon \mathbb{N}^k \to \mathbb{N}$, heißt R-berechenbar, wenn

$f(n_1, \ldots, n_k) = m$ gdw. R startet mit n_i in den Registern x_i und endet mit m im Register x_{k+1} (und Eingaben n_i in den Registern x_i)

Berechenbarkeit mit einer RM

Definition

- sei $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathrm{P})$ eine RM
- eine partielle Funktion $f \colon \mathbb{N}^k \to \mathbb{N}$, heißt *$R$-berechenbar*, wenn

  $f(n_1, \ldots, n_k) = m$ gdw. R startet mit n_i in den Registern x_i und endet mit m im Register x_{k+1} (und Eingaben n_i in den Registern x_i)
- Eine partielle Funktion $f \colon \mathbb{N}^k \to \mathbb{N}$ heißt *berechenbar auf einer RM*, wenn eine RM R existiert, sodass f R-berechenbar.

Berechenbarkeit mit einer RM

Definition

- sei $R = ((x_i)_{1 \leqslant i \leqslant n}, \mathrm{P})$ eine RM
- eine partielle Funktion $f \colon \mathbb{N}^k \to \mathbb{N}$, heißt R-berechenbar, wenn

  $f(n_1, \dots, n_k) = m$ gdw. R startet mit n_i in den Registern x_i und endet mit m im Register x_{k+1} (und Eingaben n_i in den Registern x_i)
- Eine partielle Funktion $f \colon \mathbb{N}^k \to \mathbb{N}$ heißt berechenbar auf einer RM, wenn eine RM R existiert, sodass f R-berechenbar.

Satz

Jede partielle Funktion $f \colon \mathbb{N}^k \to \mathbb{N}$, die berechenbar auf einer RM ist, ist auf einer TM berechenbar und umgekehrt

Church-Turing-These („Naturgesetz“ der Informatik)

Jedes algorithmisch lösbare Problem ist auch mit Hilfe einer Turingmaschine lösbar.

Church-Turing-These („Naturgesetz“ der Informatik)

Jedes algorithmisch lösbare Problem ist auch mit Hilfe einer Turingmaschine lösbar.

Definition (Berechenbare Reduktion)

angenommen

- L, M Sprachen über Σ
- $L \leqslant_T M$ mit $R\colon \Sigma^* \to \Sigma^*$
- die Reduktion R wird von TM T berechnet, sodass gilt

$$x \in L \Leftrightarrow R(x) \in M$$

Church-Turing-These („Naturgesetz“ der Informatik)

Jedes algorithmisch lösbare Problem ist auch mit Hilfe einer Turingmaschine lösbar.

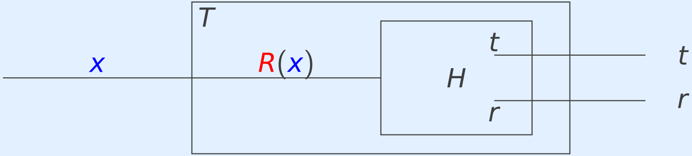
Definition (Berechenbare Reduktion)

angenommen

- L, M Sprachen über Σ
- $L \leqslant_T M$ mit $R \colon \Sigma^* \to \Sigma^*$
- die Reduktion R wird von TM T berechnet, sodass gilt

$$x \in L \Leftrightarrow R(x) \in M$$

Entscheidbarkeit von L, durch Entscheider H von M



Beispiel

Seien

$$L = \{x \in \{a,b\}^* \mid |x| \text{ ist gerade}\}$$
$$M = \{x \in \{a,b\}^* \mid x \text{ ist ein Palindrom gerader Länge}\}$$

dann gilt $L \leqslant_T M$

Beispiel

Seien

$$L = \{x \in \{a,b\}^* \mid |x| \text{ ist gerade}\}$$
$$M = \{x \in \{a,b\}^* \mid x \text{ ist ein Palindrom gerader Länge}\}$$

dann gilt $L \leqslant_T M$

Reduktion

Wir geben eine (TM) berechenbare Abbildung $R\colon \{a,b\}^* \to \{a,b\}^*$ an, sodass $x \in L \Leftrightarrow R(x) \in M$:

Beispiel

Seien

$$L = \{x \in \{a,b\}^* \mid |x| \text{ ist gerade}\}$$
$$M = \{x \in \{a,b\}^* \mid x \text{ ist ein Palindrom gerader Länge}\}$$

dann gilt $L \leqslant_T M$

Reduktion

Wir geben eine (TM) berechenbare Abbildung $R\colon \{a,b\}^* \to \{a,b\}^*$ an, sodass $x \in L \Leftrightarrow R(x) \in M$:

- definiere R', sodass $R'(a) := a$ und $R'(b) := a$
- definiere R als Erweiterung von R' auf Wörter
- R ist eine Stringfunktion, die ein Wort aus $\{a,b\}^n$ in das Wort a^n umwandelt
- Genau dann wenn n gerade ist, ist a^n ein Palindrom gerader Länge

Tabelle für $x \in L$ gdw $R(x) \in M$

$x \in L$	x	$R(x)$	$R(x) \in M$
$\checkmark$	ϵ	ϵ	$\checkmark$
$\times$	a	a	$\times$
$\times$	b	a	$\times$
$\checkmark$	aa	aa	$\checkmark$
$\checkmark$	ab	aa	$\checkmark$
$\checkmark$	ba	aa	$\checkmark$
$\checkmark$	bb	aa	$\checkmark$
$\times$	aaa	aaa	$\times$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

wobei

$L = \{x \in \{a, b\}^* \mid |x| \text{ ist gerade}\}$ $\quad M = \{x \in \{a, b\}^* \mid x \text{ ist ein Palindrom gerader Länge}\}$

Anwendungen von Reduktionen

Lemma

wenn $L \leqslant_T M$ und M entscheidbar, dann ist L entscheidbar

Anwendungen von Reduktionen

Lemma

wenn $L \leqslant_T M$ und M entscheidbar, dann ist L entscheidbar

Unentscheidbarkeit

Unentscheidbarkeit eines Problems zeigt man mittels Reduktion von einem unentscheidbares Problem (zum Beispiel das Halteproblem) auf das betrachtete Problem

Anwendungen von Reduktionen

Lemma

wenn $L \leqslant_T M$ und M entscheidbar, dann ist L entscheidbar

Unentscheidbarkeit

Unentscheidbarkeit eines Problems zeigt man mittels Reduktion von einem unentscheidbares Problem (zum Beispiel das Halteproblem) auf das betrachtete Problem

Satz

es kann kein Testprogramm für "hello, world" Programme geben

Beweis.

$$\text{HP} \leqslant_T \text{“hello, world” Programme}$$

∎