

Einführung in die Theoretische Informatik

Christian Dalvit Manuel Eberl

Samuel Frontull **Cezary Kaliszyk** Daniel Ranalter

Wintersemester 2022/23

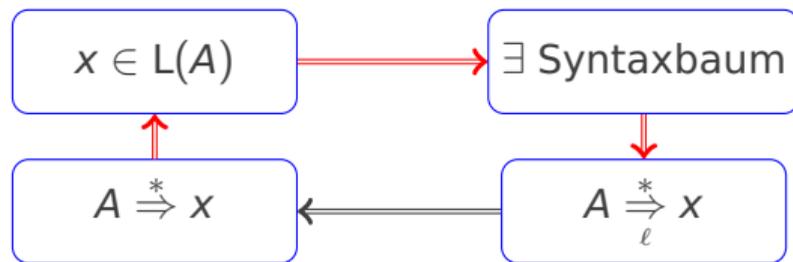
Zusammenfassung

Wintersemester 2022/23

Satz

Sei Σ ein Alphabet und sei $x \in \Sigma^*$. Die folgenden Beschreibungen kontextfreier Sprachen sind **äquivalent**:

- 1 $x \in L(A)$ nach dem rekursiven Inferenzverfahren
- 2 $A \xrightarrow{*} x$
- 3 $A \xrightarrow[\ell]{*} x$
- 4 $A \xrightarrow[r]{*} x$
- 5 Es existiert ein Syntaxbaum mit Wurzel A und Ergebnis x



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

Berechenbarkeitstheorie

Wintersemester 2022/23

Einführung in die Berechenbarkeitstheorie

Frage

Ist jedes Problem algorithmisch lösbar?

Einführung in die Berechenbarkeitstheorie

Frage

Ist jedes Problem **algorithmisch** lösbar?

Einführung in die Berechenbarkeitstheorie

Frage

Ist jedes Problem **algorithmisch** lösbar?

Antwort

Nein

Einführung in die Berechenbarkeitstheorie

Frage

Ist jedes Problem **algorithmisch** lösbar?

Antwort

Nein

Ein einfaches Programm

```
main()
{
    printf("hello, world");
}
```

Beispiel

betrachte Programm F:

```
main()
{
    int n, summe = 3, x, y, z;
    scanf("%d", &n);
    while (1) {
        for (x=1; x <= summe-2; x++)
            for (y=1; y <= summe-x-1; y++) {
                z = summe - x - y;
                if (pow(x,n) + pow(y,n) == pow(z,n))
                    printf("hello, world");
            }
        summe++;
    }
}
```

Beispiel

betrachte Programm F:

```
main()
{
    int n, summe = 3, x, y, z;
    scanf("%d", &n);
    while (1) {
        for (x=1; x <= summe-2; x++)
            for (y=1; y <= summe-x-1; y++) {
                z = summe - x - y;
                if (pow(x,n) + pow(y,n) == pow(z,n))
                    printf("hello, world");
            }
        summe++;
    }
}
```

Das Programm F ist kein „hello, world“-Programm

Beispiel (Fortsetzung)

- Code repräsentiert den großen Fermat; $x^n + y^n = z^n$, für alle n
- widerlegt von Andrew Wiles, 1980er
- In der Simpsons Folge “The Wizard of Evergreen Terrace” schreibt Homer die folgende (scheinbare) Lösung an die Tafel:

$$3987^{12} + 4365^{12} = 4472^{12} .$$

Das ist zwar falsch, aber auf einem einfachen Taschenrechner erscheint die Lösung (durch Rundungsfehler) als richtig.

Beispiel

betrachte Programm G

```
main()
{
    int n, x, y, z, test, summe=4;
    while (1) {
        test = 0;
        for (x=2; x <= summe; x++) {
            y = summe - x;
            if (is_prime(x) && is_prime(y)) test = 1;
        }
        if (!test) printf("hello, world");
        summe = summe + 2;
    }
}
```

Beispiel

betrachte Programm G

```
main()
{
    int n, x, y, z, test, summe=4;
    while (1) {
        test = 0;
        for (x=2; x <= summe; x++) {
            y = summe - x;
            if (is_prime(x) && is_prime(y)) test = 1;
        }
        if (!test) printf("hello, world");
        summe = summe + 2;
    }
}
```

Goldbach'sche Vermutung

Jede gerade Zahl größer als 2 kann als Summe zweier Primzahlen dargestellt werden

Beispiel

betrachte Programm G

```
main()
{
  int n, x, y, z, test, summe=4;
  while (1) {
    test = 0;
    for (x=2; x <= summe; x++) {
      y = summe - x;
      if (is_prime(x) && is_prime(y)) test = 1;
    }
    if (!test) printf("hello, world");
    summe = summe + 2;
  }
}
```

Das Programm G ist **wahrscheinlich** kein „hello, world“-Programm¹

¹G ist kein „hello, world“-Programm für Zahlen $\leq 10^{17}$

Großer Fermat (1637)



Die Gleichung $a^n + b^n = c^n$ besitzt für $n > 2$ keine Lösung in \mathbb{N}

Großer Fermat (1637)



Die Gleichung $a^n + b^n = c^n$ besitzt für $n > 2$ keine Lösung in \mathbb{N}

Vermutung von Goldbach (1742)



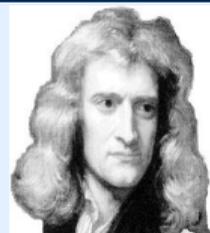
Jede gerade Zahl (≥ 2) kann als Summe zweier Primzahlen dargestellt werden

Großer Fermat (1637)



Die Gleichung $a^n + b^n = c^n$ besitzt für $n > 2$ keine Lösung in \mathbb{N}

Vermutung von Goldbach (1742)



Jede gerade Zahl (≥ 2) kann als Summe zweier Primzahlen dargestellt werden

Satz

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

Entscheidbarkeit & Unentscheidbarkeit

Definition (informell)

Ein Problem, das **algorithmisch** nicht lösbar ist, wird **unentscheidbar** genannt; andernfalls heißt das Problem **entscheidbar**

Entscheidbarkeit & Unentscheidbarkeit

Definition (informell)

Ein Problem, das **algorithmisch** nicht lösbar ist, wird **unentscheidbar** genannt; andernfalls heißt das Problem **entscheidbar**

Definition

Als **Halteproblem** bezeichnen wir das Problem, ob ein beliebiges Programm auf seiner Eingabe hält.

Entscheidbarkeit & Unentscheidbarkeit

Definition (informell)

Ein Problem, das **algorithmisch** nicht lösbar ist, wird **unentscheidbar** genannt; andernfalls heißt das Problem **entscheidbar**

Definition

Als **Halteproblem** bezeichnen wir das Problem, ob ein beliebiges Programm auf seiner Eingabe hält.

Definition

Postsches Korrespondenzproblem: Gegeben zwei gleich lange Listen von (nicht-leeren) Wörtern w_1, w_2, \dots, w_n und x_1, x_2, \dots, x_n . Gesucht sind Indizes i_1, i_2, \dots, i_m , sodass

$$w_{i_1} w_{i_2} \dots w_{i_m} = x_{i_1} x_{i_2} \dots x_{i_m}$$

Satz

Die folgenden Probleme sind *entscheidbar*:

- das Erfüllbarkeitsproblem der Aussagenlogik (SAT)

Satz

Die folgenden Probleme sind *entscheidbar*:

- das Erfüllbarkeitsproblem der Aussagenlogik (SAT)
- das Wortproblem der Booleschen Algebra

Satz

Die folgenden Probleme sind *entscheidbar*:

- das Erfüllbarkeitsproblem der Aussagenlogik (SAT)
- das Wortproblem der Booleschen Algebra
- sei G eine KFG, ist $L(G) = \emptyset$?

Satz

Die folgenden Probleme sind **entscheidbar**:

- das Erfüllbarkeitsproblem der Aussagenlogik (SAT)
- das Wortproblem der Booleschen Algebra
- sei G eine KFG, ist $L(G) = \emptyset$?
- ...

Satz

Die folgenden Probleme sind **unentscheidbar**:

- das Halteproblem
- das Postsche Korrespondenzproblem

Satz

Die folgenden Probleme sind **entscheidbar**:

- das Erfüllbarkeitsproblem der Aussagenlogik (SAT)
- das Wortproblem der Booleschen Algebra
- sei G eine KFG, ist $L(G) = \emptyset$?
- ...

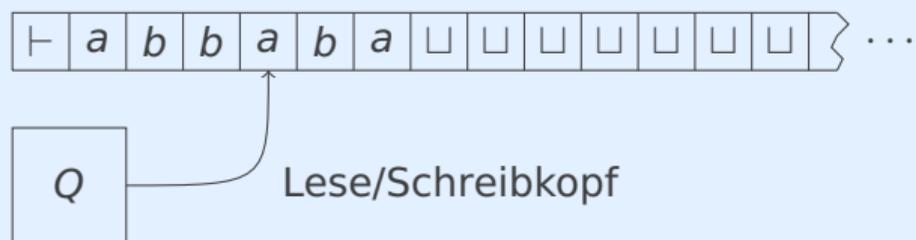
Satz

Die folgenden Probleme sind **unentscheidbar**:

- das Halteproblem
- das Postsche Korrespondenzproblem
- sei G eine KFG über Σ , ist $L(G) = \Sigma^*$?
- ...

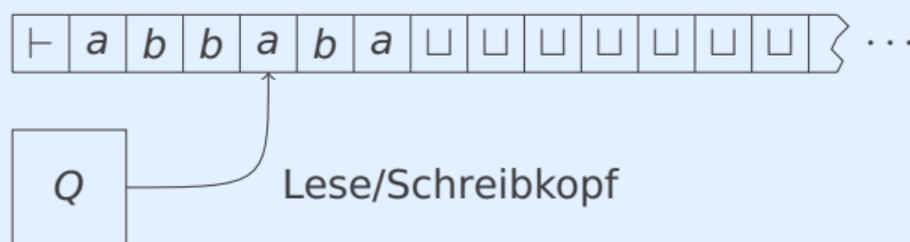
Definition (informell)

deterministische, einbändige Turingmaschine (TM):



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

Definition (formal)

eine **deterministische, einbändige Turingmaschine (TM)** M ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

Definition (formal)

eine **deterministische, einbändige Turingmaschine (TM)** M ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

- 1 Q eine endliche Menge von **Zuständen**,

Definition (formal)

eine **deterministische, einbändige Turingmaschine (TM)** M ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

- 1 Q eine endliche Menge von **Zuständen**,
- 2 Σ eine endliche Menge von **Eingabesymbolen**,

Definition (formal)

eine **deterministische, einbändige Turingmaschine (TM)** M ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

- 1 Q eine endliche Menge von **Zuständen**,
- 2 Σ eine endliche Menge von **Eingabesymbolen**,
- 3 Γ eine endliche Menge von **Bandsymbolen**, mit $\Sigma \subseteq \Gamma$,

Definition (formal)

eine **deterministische, einbändige Turingmaschine (TM)** M ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

- 1 Q eine endliche Menge von **Zuständen**,
- 2 Σ eine endliche Menge von **Eingabesymbolen**,
- 3 Γ eine endliche Menge von **Bandsymbolen**, mit $\Sigma \subseteq \Gamma$,
- 4 $\vdash \in \Gamma \setminus \Sigma$, der **linke Endmarker**,

Definition (formal)

eine **deterministische, einbändige Turingmaschine (TM)** M ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

- 1 Q eine endliche Menge von **Zuständen**,
- 2 Σ eine endliche Menge von **Eingabesymbolen**,
- 3 Γ eine endliche Menge von **Bandsymbolen**, mit $\Sigma \subseteq \Gamma$,
- 4 $\vdash \in \Gamma \setminus \Sigma$, der **linke Endmarker**,
- 5 $\sqcup \in \Gamma \setminus \Sigma$, das **Blanksymbol**,

Definition (formal)

eine **deterministische, einbändige Turingmaschine (TM)** M ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

- 1 Q eine endliche Menge von **Zuständen**,
- 2 Σ eine endliche Menge von **Eingabesymbolen**,
- 3 Γ eine endliche Menge von **Bandsymbolen**, mit $\Sigma \subseteq \Gamma$,
- 4 $\vdash \in \Gamma \setminus \Sigma$, der **linke Endmarker**,
- 5 $\sqcup \in \Gamma \setminus \Sigma$, das **Blanksymbol**,
- 6 $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ die **Übergangsfunktion**,

Definition (formal)

eine **deterministische, einbändige Turingmaschine (TM)** M ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

- 1 Q eine endliche Menge von **Zuständen**,
- 2 Σ eine endliche Menge von **Eingabesymbolen**,
- 3 Γ eine endliche Menge von **Bandsymbolen**, mit $\Sigma \subseteq \Gamma$,
- 4 $\vdash \in \Gamma \setminus \Sigma$, der **linke Endmarker**,
- 5 $\sqcup \in \Gamma \setminus \Sigma$, das **Blanksymbol**,
- 6 $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ die **Übergangsfunktion**,
- 7 $s \in Q$, der **Startzustand**,

Definition (formal)

eine **deterministische, einbändige Turingmaschine (TM)** M ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

- 1 Q eine endliche Menge von **Zuständen**,
- 2 Σ eine endliche Menge von **Eingabesymbolen**,
- 3 Γ eine endliche Menge von **Bandsymbolen**, mit $\Sigma \subseteq \Gamma$,
- 4 $\vdash \in \Gamma \setminus \Sigma$, der **linke Endmarker**,
- 5 $\sqcup \in \Gamma \setminus \Sigma$, das **Blanksymbol**,
- 6 $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ die **Übergangsfunktion**,
- 7 $s \in Q$, der **Startzustand**,
- 8 $t \in Q$, der **akzeptierende Zustand** und

Definition (formal)

eine **deterministische, einbändige Turingmaschine (TM)** M ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, \vdash, \sqcup, \delta, s, t, r)$$

sodass

- 1 Q eine endliche Menge von **Zuständen**,
- 2 Σ eine endliche Menge von **Eingabesymbolen**,
- 3 Γ eine endliche Menge von **Bandsymbolen**, mit $\Sigma \subseteq \Gamma$,
- 4 $\vdash \in \Gamma \setminus \Sigma$, der **linke Endmarker**,
- 5 $\sqcup \in \Gamma \setminus \Sigma$, das **Blanksymbol**,
- 6 $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ die **Übergangsfunktion**,
- 7 $s \in Q$, der **Startzustand**,
- 8 $t \in Q$, der **akzeptierende Zustand** und
- 9 $r \in Q$, der **verwerfende Zustand** mit $t \neq r$.

Definition (Übergangsfunktion)

die Gleichung $\delta(p, a) = (q, b, d)$ bedeutet: Wenn die TM M im Zustand p das Symbol a liest, dann

- 1 M ersetzt a durch b auf dem Band
- 2 der Lese/Schreibkopf bewegt sich einen Schritt in die Richtung d
- 3 M wechselt in den Zustand q

Definition (Übergangsfunktion)

die Gleichung $\delta(p, a) = (q, b, d)$ bedeutet: Wenn die TM M im Zustand p das Symbol a liest, dann

- 1 M ersetzt a durch b auf dem Band
- 2 der Lese/Schreibkopf bewegt sich einen Schritt in die Richtung d
- 3 M wechselt in den Zustand q

Definition (Zusatzbedingungen)

- Der linke Endmarker darf nicht überschrieben werden

$$\forall p \in Q, \exists q \in Q \quad \delta(p, \vdash) = (q, \vdash, R)$$

Definition (Übergangsfunktion)

die Gleichung $\delta(p, a) = (q, b, d)$ bedeutet: Wenn die TM M im Zustand p das Symbol a liest, dann

- 1 M ersetzt a durch b auf dem Band
- 2 der Lese/Schreibkopf bewegt sich einen Schritt in die Richtung d
- 3 M wechselt in den Zustand q

Definition (Zusatzbedingungen)

- Der linke Endmarker darf nicht überschrieben werden

$$\forall p \in Q, \exists q \in Q \quad \delta(p, \vdash) = (q, \vdash, R)$$

- Wenn die TM akzeptiert/verwirft, bleibt die TM in diesem Zustand

$$\forall b \in \Gamma, \exists c, c' \in \Gamma \text{ und } d, d' \in \{L, R\}: \begin{aligned} \delta(t, b) &= (t, c, d) \\ \delta(r, b) &= (r, c', d') \end{aligned}$$

Beispiel

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

Beispiel

sei $M = (\{s, t, r, q_1, q_2, q_3\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1, X, Y\}, \vdash, \sqcup, \delta, s, t, r)$ 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, R)			
s	0	(q_1, X, R)	q_2	0	$(q_2, 0, L)$
s	Y	(q_3, Y, R)	q_2	X	(s, X, R)
			q_2	Y	(q_2, Y, L)
			q_3	\sqcup	(t, \sqcup, R)
q_1	0	$(q_1, 0, R)$			
q_1	1	(q_2, Y, L)			
q_1	Y	(q_1, Y, R)	q_3	Y	(q_3, Y, R)

Beispiel

sei $M = (\{s, t, r, q_1, q_2, q_3\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1, X, Y\}, \vdash, \sqcup, \delta, s, t, r)$ 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, R)			
s	0	(q_1, X, R)	q_2	0	$(q_2, 0, L)$
s	Y	(q_3, Y, R)	q_2	X	(s, X, R)
			q_2	Y	(q_2, Y, L)
			q_3	\sqcup	(t, \sqcup, R)
q_1	0	$(q_1, 0, R)$			
q_1	1	(q_2, Y, L)			
q_1	Y	(q_1, Y, R)	q_3	Y	(q_3, Y, R)
t	$*$	$(t, *, R)$	r	$*$	$(r, *, R)$

Beispiel

sei $M = (\{s, t, r, q_1, q_2, q_3\}, \{0, 1\}, \{\vdash, \sqcup, 0, 1, X, Y\}, \vdash, \sqcup, \delta, s, t, r)$ 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, R)	q_2	\vdash	(r, \vdash, R)
s	\sqcup	(r, \sqcup, R)	q_2	\sqcup	(r, \sqcup, R)
s	0	(q_1, X, R)	q_2	0	$(q_2, 0, L)$
s	1	$(r, 1, R)$	q_2	1	$(r, 1, R)$
s	X	(r, X, R)	q_2	X	(s, X, R)
s	Y	(q_3, Y, R)	q_2	Y	(q_2, Y, L)
q_1	\vdash	(r, \vdash, R)	q_3	\vdash	(r, \vdash, R)
q_1	\sqcup	(r, \vdash, R)	q_3	\sqcup	(t, \sqcup, R)
q_1	0	$(q_1, 0, R)$	q_3	0	$(r, 0, R)$
q_1	1	(q_2, Y, L)	q_3	1	$(r, 1, R)$
q_1	X	(r, X, R)	q_3	X	(r, X, R)
q_1	Y	(q_1, Y, R)	q_3	Y	(q_3, Y, R)
t	$*$	$(t, *, R)$	r	$*$	$(r, *, R)$

Konfiguration einer TM

Definition

eine **Konfiguration** einer TM M ist ein Tripel (p, x, n) , sodass

- 1 $p \in Q$ Zustand,
- 2 $x = y \sqcup^{\infty}$ Bandinhalt
- 3 $n \in \mathbb{N}$ Position des Lese/Schreibkopfes

$y \in \Gamma^*$

Konfiguration einer TM

Definition

eine **Konfiguration** einer TM M ist ein Tripel (p, x, n) , sodass

- 1 $p \in Q$ Zustand,
- 2 $x = y \sqcup^\infty$ Bandinhalt
- 3 $n \in \mathbb{N}$ Position des Lese/Schreibkopfes

$y \in \Gamma^*$

Definition

Startkonfiguration bei Eingabe $x \in \Sigma^*$: $(s, \vdash x \sqcup^\infty, 0)$

Konfiguration einer TM

Definition

eine **Konfiguration** einer TM M ist ein Tripel (p, x, n) , sodass

- 1 $p \in Q$ Zustand,
- 2 $x = y \sqcup^\infty$ Bandinhalt
- 3 $n \in \mathbb{N}$ Position des Lese/Schreibkopfes

$y \in \Gamma^*$

Definition

Startkonfiguration bei Eingabe $x \in \Sigma^*$: $(s, \vdash x \sqcup^\infty, 0)$

Beispiel

Sei 0011 die Eingabe der TM M , dann ist die Startkonfiguration gegeben durch $(s, \vdash 0011 \sqcup^\infty, 0)$

Schrittfunktion von TMs

Definition

Relation $\xrightarrow[M]{1}$ ist wie folgt definiert:

$$(p, z, n) \xrightarrow[M]{1} \begin{cases} (q, z', n - 1) & \text{wenn } \delta(p, z_n) = (q, b, L) \\ (q, z', n + 1) & \text{wenn } \delta(p, z_n) = (q, b, R) \end{cases}$$

z' ist String, den wir aus z erhalten, wenn z_n durch b ersetzt

Schrittfunktion von TMs

Definition

Relation $\xrightarrow[M]{1}$ ist wie folgt definiert:

$$(p, z, n) \xrightarrow[M]{1} \begin{cases} (q, z', n - 1) & \text{wenn } \delta(p, z_n) = (q, b, L) \\ (q, z', n + 1) & \text{wenn } \delta(p, z_n) = (q, b, R) \end{cases}$$

z' ist String, den wir aus z erhalten, wenn z_n durch b ersetzt

Definition

reflexive, transitive Hülle $\xrightarrow[M]{*}$:

Schrittfunktion von TMs

Definition

Relation $\xrightarrow[M]{1}$ ist wie folgt definiert:

$$(p, z, n) \xrightarrow[M]{1} \begin{cases} (q, z', n - 1) & \text{wenn } \delta(p, z_n) = (q, b, L) \\ (q, z', n + 1) & \text{wenn } \delta(p, z_n) = (q, b, R) \end{cases}$$

z' ist String, den wir aus z erhalten, wenn z_n durch b ersetzt

Definition

reflexive, transitive Hülle $\xrightarrow[M]{*}$:

$$\mathbf{1} \quad \alpha \xrightarrow[M]{0} \alpha$$

Schrittfunktion von TMs

Definition

Relation $\xrightarrow[M]{1}$ ist wie folgt definiert:

$$(p, z, n) \xrightarrow[M]{1} \begin{cases} (q, z', n-1) & \text{wenn } \delta(p, z_n) = (q, b, L) \\ (q, z', n+1) & \text{wenn } \delta(p, z_n) = (q, b, R) \end{cases}$$

z' ist String, den wir aus z erhalten, wenn z_n durch b ersetzt

Definition

reflexive, transitive Hülle $\xrightarrow[M]{*}$:

1 $\alpha \xrightarrow[M]{0} \alpha$

2 $\alpha \xrightarrow[M]{n+1} \beta$, wenn $\alpha \xrightarrow[M]{n} \gamma \xrightarrow[M]{1} \beta$ für Konfiguration γ

Schrittfunktion von TMs

Definition

Relation $\xrightarrow[M]{1}$ ist wie folgt definiert:

$$(p, z, n) \xrightarrow[M]{1} \begin{cases} (q, z', n-1) & \text{wenn } \delta(p, z_n) = (q, b, L) \\ (q, z', n+1) & \text{wenn } \delta(p, z_n) = (q, b, R) \end{cases}$$

z' ist String, den wir aus z erhalten, wenn z_n durch b ersetzt

Definition

reflexive, transitive Hülle $\xrightarrow[M]{*}$:

1 $\alpha \xrightarrow[M]{0} \alpha$

2 $\alpha \xrightarrow[M]{n+1} \beta$, wenn $\alpha \xrightarrow[M]{n} \gamma \xrightarrow[M]{1} \beta$ für Konfiguration γ

3 $\alpha \xrightarrow[M]{*} \beta$, wenn $\exists n \alpha \xrightarrow[M]{n} \beta$