VL-02: Turing Maschinen II

(Berechenbarkeit und Komplexität, WS 2018)

Gerhard Woeginger

WS 2018, RWTH

Organisatorisches

- Nächste Vorlesung (morgen):
  Freitag, Oktober 26, 16:30–18:00 Uhr, Audimax
- Webseite:
  http://algo.rwth-aachen.de/Lehre/WS1819/BuK.php

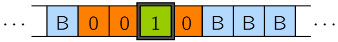
Wiederholung

Wdh.: Kodierung von Berechnungsproblemen

Drei mögliche formale Definitionen:

Als Relation:	Als Funktion:	Als Sprache:
Primfaktor:		
$(110, 11) \in R$		
$(101, 11) \notin R$		
$(00110, 11) \notin R$		
Multiplikation:		
$(11\#10, 110) \in R$	$f(11\#10) = 110$	
$(11\#10, 11) \notin R$		
$(1\#1\#0, 110) \notin R$	$(f(1\#1\#0) = \perp)$	
Wörter die auf 1 enden:		
$(11, 1) \in R$	$f(11) = 1$	$11 \in L$
$(110, 1) \notin R$	$f(110) = 0$	$110 \notin L$
$(10, 0) \in R$		

Anschauliche Definition:



δ	0	1	B
q_1	$(q_1, 1, L)$	$(q_2, 1, R)$	(q_1, B, N)
q_2	(q_3, B, R)	$(q_1, 0, L)$	(q_3, B, R)
q_3	$(q_2, 0, N)$	$(q_2, 0, R)$	(q_3, B, R)

Formale Definition:

Eine Turingmaschine ist ein 7-Tupel $(Q, \Sigma, \Gamma, B, q_0, \bar{q}, \delta)$, wobei

- Q, Σ, Γ endliche Mengen sind,
- $\Sigma \subseteq \Gamma$,
- $B \in \Gamma \setminus \Sigma$,
- $q_0, \bar{q} \in Q$ und
- $\delta \colon (Q \setminus \{\bar{q}\}) \times \Gamma \to Q \times \Gamma \times \{R, L, N\}$.

Wdh.: Funktionsweise der TM (1)

Ausgangssituation

- Auf dem Band steht die Eingabe $x \in \Sigma^*$ eingerahmt von Blanks
- Der initiale Zustand ist q_0
- Der Kopf steht über dem ersten Symbol von x

Durchführung eines einzelnen Rechenschrittes

- $a \in \Gamma$ bezeichne das gelesene Symbol (unterm Kopf)
- $q \in Q \setminus \{\bar{q}\}$ bezeichne den aktuellen Zustand
- Angenommen $\delta(q, a) = (q', a', d)$, für $q' \in Q, a' \in \Gamma, d \in \{R, L, N\}$

- Dann wird der Zustand auf q' gesetzt
- An der Kopfposition wird das Symbol a' geschrieben
- Der Kopf bewegt sich um eine Position nach rechts (falls $d = R$), oder um eine Position nach links (falls $d = L$), oder er bewegt sich gar nicht (falls $d = N$)

Wdh.: Funktionsweise der TM (2)

Ende der Berechnung

- Die TM stoppt, wenn sie den Endzustand $\bar{q}$ erreicht
- Das Ausgabewort $y \in \Sigma^*$ kann dann vom Band abgelesen werden: y beginnt an der Kopfposition und endet unmittelbar vor dem ersten Symbol aus $\Gamma \setminus \Sigma$

Ende der Berechnung / Variante

- Auf dem Band steht die Ausgabe $y \in \Sigma^*$ eingerahmt von Blanks
- Der aktuelle Zustand ist der Endzustand $\bar{q}$
- Der Kopf steht über dem ersten Symbol von y

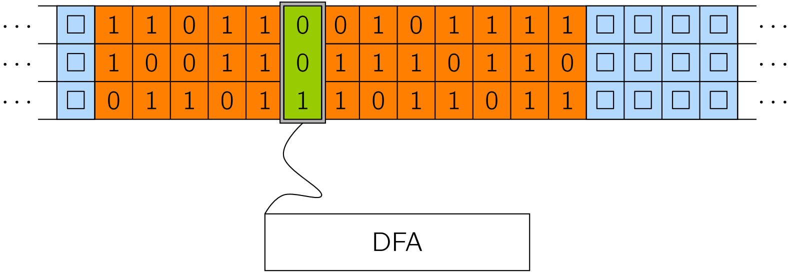
Frage: Macht diese Variante das TM Modell stärker oder schwächer?

Wdh.: Laufzeit und Speicherbedarf

- Laufzeit = Anzahl von Zustandsübergängen bis zur Terminierung
- Speicherbedarf = Anzahl von Bandzellen, die während der Berechnung besucht werden

Wdh.: TM-Techniken

- Speicher im Zustandsraum: $Q_{\text{neu}} := Q \times \Gamma^k$
- Mehrspurmaschinen: $\Gamma_{\text{neu}} := \Gamma \cup \Gamma^k$



- Schleifen, Variablen, Felder (Arrays), Unterprogramme

Vorlesung VL-02
Turing Maschinen II

- Mehrband-Turingmaschinen
- Gödelnummern
- Die universelle Turingmaschine

- Church-Turing These

Mehrband-Turingmaschinen

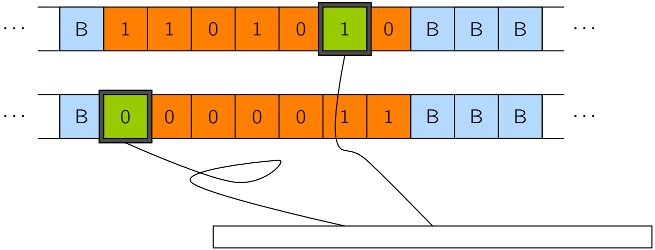
Turingmaschinen mit mehreren Bändern

k-Band-TM

Eine k-Band-TM ist eine Verallgemeinerung der Turingmaschine.
Sie verfügt über k Arbeitsbänder mit jeweils einem unabhängigen Kopf.
Die Zustandsübergangsfunktion ist entsprechend von der Form

$$\delta : (Q \setminus \{\bar{q}\}) \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, N\}^k \ .$$

- Band 1 fungiert als Ein-/Ausgabeband wie bei der (1-Band-)TM.
- Die Zellen der Bänder $2, \ldots, k$ sind anfangs leer (ausschliesslich B).



Simulation k-Band-TM durch 1-Band-TM (1)

Satz

Eine k-Band-TM M mit Zeitbedarf $t(n)$ und Platzbedarf $s(n)$ kann von einer 1-Band-TM M' mit Zeitbedarf $O(t^2(n))$ und Platzbedarf $O(s(n))$ simuliert werden.

Beweisskizze:
Die TM M' verwendet $2k$ Spuren.
Nach Simulation des t-ten Schrittes für $0 \leq t \leq t(n)$ gilt

- Die ungeraden Spuren $1, 3, \ldots, 2k-1$ enthalten den Inhalt der Bänder $1, \ldots, k$ von M.
- Auf den geraden Spuren $2, 4, \ldots, 2k$ sind die Kopfpositionen auf diesen Bändern mit dem Zeichen $\#$ markiert.

Simulation k-Band-TM durch 1-Band-TM (2)

Simulierte 2-Band-Turingmaschine M:

…	B	*	*	*	*	A	r	b	e	i	t	s	b	a	n	d	1	B	B	B	…

…	B	A	r	b	e	i	t	s	b	a	n	d	2	*	*	*	*	B	B	B	…

Simulierende 4-Spur-Turingmaschine M'
(zu Beginn des Simulationsschrittes):

…	B	*	*	*	*	A	r	b	e	i	t	s	b	a	n	d	1	B	B	B	…
…	B	B	B	B	B	B	#	B	B	B	B	B	B	B	B	B	B	B	B	B	…
…	B	A	r	b	e	i	t	s	b	a	n	d	2	*	*	*	*	B	B	B	…
…	B	#	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	…

Simulation k-Band-TM durch 1-Band-TM (3)

Jeder Rechenschritt von M wird durch M' wie folgt simuliert.

- Am Anfang steht der Kopf von M' auf der linkesten Zelle, die $\#$ enthält, und M' kennt den Zustand von M.
- Der Kopf von M' läuft dann nach rechts bis zum rechtesten $\#$, wobei die k Zeichen an den mit $\#$ markierten Spurpositionen im Zustand abgespeichert werden.
- An der Zelle mit dem rechtesten $\#$-Zeichen angekommen, kann M' die Übergangsfunktion von M auswerten und kennt den neuen Zustand von M sowie die erforderlichen Übergänge auf den k Bändern.
- Nun läuft der Kopf von M' zurück, verändert dabei die Bandinschriften an den mit $\#$ markierten Stellen und verschiebt, falls erforderlich, auch die $\#$-Markierungen um eine Position nach links oder rechts.

Simulation k-Band-TM durch 1-Band-TM (4)

Laufzeitanalyse:

- Wieviele Bandpositionen können zwischen dem linkesten und dem rechtesten $\#$ liegen?
- Nach t Schritten von M können diese Markierungen höchstens $2t$ Positionen auseinanderliegen.
- Also ist der Abstand zwischen diesen Zeichen und somit auch die Laufzeit zur Simulation eines einzelnen Schrittes durch $O(t(n))$ beschränkt.
- Insgesamt ergibt das dann für die Simulation von $t(n)$ Schritten eine Laufzeitschranke von $O(t(n)^2)$. Q.E.D.

Die universelle Turingmaschine

Special Purpose versus general Purpose Rechner

- Bisher haben wir für jedes Problem eine eigene TM entworfen, einen special purpose Rechner.
- Real existierende Maschinen sind jedoch programmierbare general purpose Rechner.
- Wir konstruieren jetzt eine programmierbare Variante der TM, die sogenannte universelle TM.

Ein-/Ausgabeverhalten der universellen TM

- Das "Programm" der universellen TM U ist die Kodierung einer beliebigen TM M.
- Mit $\langle M\rangle$ bezeichnen wir diese Kodierung der TM M.
- Als Eingabe erhält U einen String der Form $\langle M\rangle w$ bestehend aus einer TM-Kodierung $\langle M\rangle$ und einem Wort w.
- Die universelle TM simuliert das Verhalten der TM M auf der Eingabe w.
- Bei inkorrekter Eingabe (z.B.: wenn die Eingabe nicht mit einer TM-Kodierung beginnt; oder: wenn das Wort w Buchstaben enthält, die nicht zum Eingabealphabet von M gehören) gibt U eine Fehlermeldung aus.

Gödelnummern (1)

Wir entwickeln nun eine eindeutige präfixfreie Kodierung, die jeder Turingmaschine M ein Wort $\langle M \rangle$ über dem Alphabet $\{0, 1\}$ zuordnet.

Definition

Wir nennen die Kodierung $\langle M \rangle$ die Gödelnummer der Turingmaschine M.

- *Präfixfrei* bedeutet, dass keine Gödelnummer Präfix (Anfangsteilwort) einer anderen Gödelnummer sein darf.
- Um Präfixfreiheit zu erreichen, vereinbaren wir, dass alle Gödelnummern mit 111 beginnen und auf 111 enden und ansonsten den Teilstring 111 nicht in ihrer Kodierung enthalten.

Gödelnummern (2)

Zur präfixfreien Kodierung von TMen gibt es viele Möglichkeiten.
Wir stellen jetzt eine mögliche Definition der Gödelnummer vor.

Wir beschränken uns auf TMen der folgenden Form:

- $Q = \{q_1, \ldots, q_t\}$ für ein $t \geq 2$.
- Der Anfangszustand ist q_1 und der Endzustand ist q_2.
- $\Gamma = \{0, 1, B\}$.

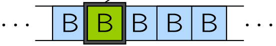
Zur Beschreibung von TMen dieser Form müssen wir nur die Übergangsfunktion als Binärstring kodieren.

- Wir nummerieren das Alphabet durch, indem wir $X_1 = 0$, $X_2 = 1$ und $X_3 = B$ setzen.
- Auch die möglichen Kopfbewegungen nummerieren wir, indem wir $D_1 = L$, $D_2 = N$ und $D_3 = R$ setzen.

Gödelnummern (3)

Kodierung der Übergangsfunktion

- Der Übergang $\delta(q_i, X_j) = (q_k, X_\ell, D_m)$ wird kodiert durch den Binärstring
$$0^i 1 0^j 1 0^k 1 0^\ell 1 0^m$$
- Die Kodierung des t-ten Übergangs bezeichnen wir mit $\mathrm{code}(t)$.
- Die Gödelnummer einer TM M mit s Übergängen ist dann
$$\langle M \rangle \;=\; 111\,\mathrm{code}(1)\,11\,\mathrm{code}(2)\,11 \ldots 11\,\mathrm{code}(s)\,111$$



$\Sigma = \{0, 1\}$
$\Gamma = \{0, 1, B\}$
$Q = \{q_1, q_2, q_3\}$

Start: q_1 Ende: q_2 Blank: B

δ	0	1	B
q_1	(q_1, B, R)	(q_3, B, R)	(q_2, B, N)
q_3	$(q_1, 1, R)$	$(q_2, 0, R)$	(q_1, B, L)

111 0101010001000 11 0100100010001000 11 010001001000100 11
000101010010000 11 000100100101000 11 000100010100010 111

Implementierung der universellen TM (1)

Als Eingabe erhält die universelle TM U
ein Wort der Form $\langle M \rangle w$ für beliebiges $w \in \{0, 1\}^*$.

Wir implementieren U zunächst in Form einer 3-Band-TM:

- Band 1 von U simuliert das Band der TM M.
- Band 2 von U enthält die Gödelnummer von M.
- Auf Band 3 speichert U den jeweils aktuellen Zustand von M.

Implementierung der universellen TM (2a)

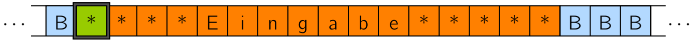
Initialisierung:

- U überprüft, ob die Eingabe eine korrekte Gödelnummer enthält. Falls nein: Fehlermeldung.
- U kopiert die Gödelnummer auf Band 2 und schreibt die Kodierung des Anfangszustands auf Band 3.
- U bereitet Band 1 so vor, dass es nur das Wort w enthält. Der Kopf steht auf dem ersten Zeichen von w.

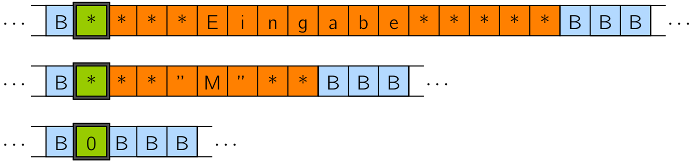
Die Laufzeit der Initialisierung ist $O(1)$,
wenn wir die Kodierungslänge von M als Konstante ansehen.

Implementierung der universellen TM (2b)

Simulierte Turingmaschine *M*



Initialisierung der universellen Maschine *U*



Implementierung der universellen TM (3)

Simulation eines einzelnen Schritts von M:

U sucht zu dem Zeichen an der Kopfposition auf Band 1 und zu dem Zustand auf Band 3 die Kodierung des entsprechenden Übergangs von M auf Band 2.

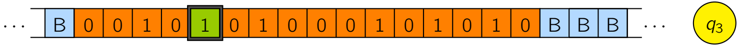
Wie in der Übergangsfunktion beschrieben

- aktualisiert U die Inschrift auf Band 1,
- bewegt U den Kopf auf Band 1, und
- verändert U den auf Band 3 abgespeicherten Zustand von M.

Laufzeit eines Simulationsschrittes: $O(1)$.
U simuliert M mit konstantem Zeitverlust!

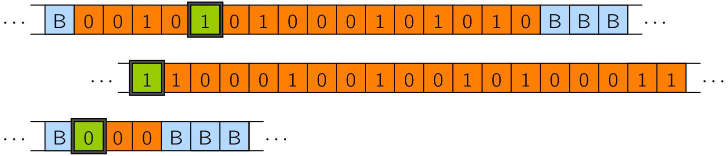
Simulation durch universelle Maschine: Illustration

Simulierte Turingmaschine M



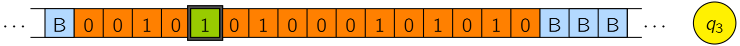
δ	0	1	B
q_1			
q_2			
q_3		$(q_2, 0, R)$	

Simulierende universelle Maschine U



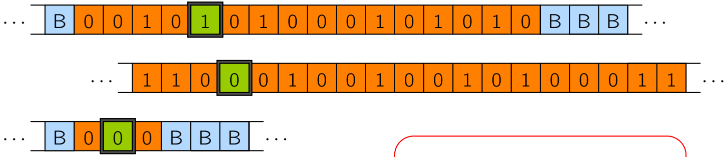
Simulation durch universelle Maschine: Illustration

Simulierte Turingmaschine M



δ	0	1	B
q_1			
q_2			
q_3		$(q_2, 0, R)$	

Simulierende universelle Maschine U



Vergleiche Band 3 mit Band 2

Implementierung der universellen TM (4)

- Können wir dieses Ergebnis auch mit einer (1-Band-)TM erreichen?
- Natürlich können wir die beschriebene 3-Band-TM auf der 1-Band TM mit mehreren Spuren simulieren.
- Aber bei Verwendung dieser Simulation handeln wir uns einen quadratischen Zeitverlust ein.
- Wir erhalten eine universelle 1-Band-TM mit konstantem Zeitverlust, wenn wir die Gödelnummer auf Spur 2 und den Zustand auf Spur 3 mit dem Kopf der TM M mitführen.

Alan Mathison Turing OBE FRS (1912–1954)

Wikipedia: Alan Turing was an English computer scientist, mathematician, logician, cryptanalyst, philosopher and theoretical biologist.

Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general purpose computer. Turing is widely considered to be the father of theoretical computer science and artificial intelligence.

Die Church-Turing These

Alonzo Church (1903–1995)

Wikipedia: Alonzo Church was an American mathematician and logician who made major contributions to mathematical logic and the foundations of theoretical computer science.

He is best known for the creation of the lambda calculus, the Church-Turing thesis, proving that the Entscheidungsproblem is undecidable, proving that Peano arithmetic is undecidable, Frege-Church ontology, and the Church-Rosser theorem. Alonzo Church was the founding editor of the Journal of Symbolic Logic, and he edited its reviews section until 1979.

Die Church-Turing These

Kein bekanntes "vernünftiges" Rechnermodell ist mächtiger als die TM:

- Lambda-Kalkül von Alonzo Church: gleich mächtig wie TM
- Kanonische Systeme von Emil Post: gleich mächtig wie TM
- μ-rekursive Funktionen von Kurt Gödel: gleich mächtig wie TM

Alonzo Church und Alan Turing formulierten in den 1930er Jahren:

Church-Turing These

Die Klasse der TM-berechenbaren Funktionen stimmt mit der Klasse der "intuitiv berechenbaren" Funktionen überein.

Sprachliche Übereinkunft für den Rest der Vorlesung:
berechenbare Funktion= TM-berechenbare Funktion= rekursive Funktion
entscheidbare Sprache= TM-entscheidbare Sprache= rekursive Sprache