SS 2015

Zentralübung zur Vorlesung Theoretische Informatik

Dr. Werner Meixner

Fakultät für Informatik
TU München

http://www14.in.tum.de/lehre/2015SS/theo/uebung/

18. Juni 2015

ZÜ VII

Übersicht:

1. Übungsbetrieb	Klausur: Termin, Ort, Anmeldung, Ablauf, Code
2. Thema	Abstrakte Automaten und Maschinen Beispiel
3. Vorbereitung	TA Blatt 9

1. Übungsbetrieb

1.1 Klausur

Fragen zum Stoff der Endterm?

- Klausurergebnisse
- Klausureinsicht

1.2 Termin und Ort

Zeit: Donnerstag, 30. Juli, 11 – 14 Uhr

Ort: Hörsäle MW 2001 (+Galerie)
MI HS1, Interim HS1.

Bitte mindestens 15 Minuten vor Beginn im Hörsaal erscheinen!

Platzverteilung:
Die Zuordnung der Teilnehmer auf die Hörsäle erfolgt nach Abschnitten des Alphabets siehe Übungswebseite ab 27. Juli.

Die Verteilung auf Sitzplätze wird den Listen zu entnehmen sein, die an den Hörsaaleingängen aushängen werden.

1.3 Anmeldung

Eine Anmeldung für Endterm bzw. Wiederholungsklausur erfolgt über TUMonline (läuft derzeit) oder in Sonderfällen persönlich am Infopoint.

Achtung:

Bei Nichtanmeldung kann nicht garantiert werden, dass Sitzplatz und Klausurunterlagen zur Verfügung stehen!

Alle Teilnehmer der Klausuren müssen sich bei der Ausweiskontrolle im Hörsaal ausweisen können!!

1.4 Ablauf

Es nicht erlaubt, Unterlagen zu benutzen,
außer
einem persönlich handbeschriebenen DIN A4 Blatt (beidseitig)

Fragen während der Klausur sind erlaubt,
aber
Antworten werden, falls notwendig,
nur als Hörsaalansage gegeben.

1.5 Code

Code:								

Bitte 8-stelligen Zifferncode (nicht ausschließlich die Null) eintragen,
falls Sie Ihr Ergebnis frühzeitig erfahren wollen.

1.6 Fragen, Anregungen?

Aktuelle Fragen?

2. Thema Abstrakte Automaten und Maschinen

Automaten sind gedachte mathematische Objekte, die grundsätzlich aus folgenden Komponenten bestehen:

- Eingabeeinheit
- Steuereinheit
- Speichereinheit

Maschinen haben zusätzlich eine

- Ausgabeeinheit

Eingabeeinheit

Eingabeband Felder zur Aufnahme genau je eines Zeichens des Eingabealphabets, Sonderzeichen

Lesekopf

Konfiguration der Eingabeeinheit:
Eingabewort, Stellung des Lesekopfes

Konfigurationsänderung
durch Bewegung des Lesekopfes nach links oder rechts, bzw. Eingabebandes nach rechts oder links, einseitig oder zweiseitig

Steuereinheit

Zustände	stets ist genau ein Zustand aktiv
Programm	endlicher Text, definiert die möglichen Konfigurations- und Zustandsübergänge

Speichereinheit

Speicherbänder: Felder zur Aufnahme genau je eines Zeichens des Speicheralphabets, Sonderzeichen

Lese-Schreibköpfe

Konfiguration der Speichereinheit:
Speicherwörter,
Stellung der Lese-Schreibköpfe

Konfigurationsänderung
durch Bewegung der Lese-Schreibköpfe nach links oder rechts, bzw. Speicherbänder nach rechts oder links, Schreiben von Zeichen

Ausgabeeinheit

Ausgabeband: Felder zur Aufnahme genau je eines Zeichens des Ausgabealphabets, Sonderzeichen

Schreibköpfe

Konfiguration der Ausgabeeinheit:
Ausgabewort, Stellung des Schreibkopfes

Konfigurationsänderung
durch Bewegung des Schreibkopfes nach links oder rechts, bzw. Ausgabebandes nach rechts oder links, Schreiben von Zeichen

Zustandsbeschreibung, Situation

Zu jedem Zeitpunkt des Arbeitens eines Automaten oder einer Maschine M gilt eine Zustandsbeschreibung bestehend aus allen Konfigurationen der Einheiten und dem momentanen Zustand der Steuereinheit.

Eine Zustandsbeschreibung nennt man auch Situation von M.

Startsituation

Es wird definiert, mit welchen Konfigurationen und in welchem Zustand eine Maschine gestartet wird.

Berechnung

Das Programm definiert die mögliche Aufeinanderfolge der Situationen, d.h. die Einzelübergänge.

Sie definiert eine Relation $\vdash_M$ zwischen den Situationen.

Eine definierte Situationsfolge nennt man Berechnung.

Ein einzelner Übergang kann
deterministisch oder nichtdeterministisch erfolgen.

Berechnungsende, Halt

Falls kein zulässiger Situationsübergang definiert ist, hält M. Beispiel: leerer Keller.

2.1 Beispiele

Alle konkreten Automaten und Maschinen ergeben sich als Kombination der grundsätzlich möglichen Komponenten.

In der Vorlesung und den Übungen werden folgende Automaten und Maschinen besprochen:

Endliche Automaten:	DFA, NFA, ϵ-NFA, ϵ-DFA
Deterministische Kellerautomaten:	DPDA
Kellerautomaten:	PDA
2-Kellerautomat:	2KA, 2PDA
Queue-Automaten:	QA
Turingmaschinen:	DTM, NTM,
Linear beschränkte Automaten:	LBA
Mehrband Turingmaschinen:	MBTM

Gerät	Spezifikation	Konfiguration
EA:	$A = (Q, \Sigma, \delta, S, F)$	(q, w)
DPDA:	$K = (Q, \Sigma, \Delta, q_0, Z_0, \delta, F)$	(q, w, α)
PDA:	wie DPDA	
2-KA:	$K = (Q, \Sigma, \Delta, q_0, Z_0, Z_0', \delta, F)$	$(q, w, \alpha_1, \alpha_2)$
QA:	$QA = (Q, \Sigma, \Gamma, q_0, Z_0, \delta)$	(q, α, β)
TM:	$T = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$	(α, q, β)
LBA:	wie TM	
MBTM:	$T = (Q, \Sigma, \Gamma, \delta, q_0, \square, F)$	
	mit $\delta : Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, N\}^k$	$(\vec{\alpha}, q, \vec{\beta})$

Alle Spezifikationen besitzen auch Darstellungen durch Graphen.

2.2 Sätze

Sätze beziehen sich auf die

1. Spezifikationseigenarten: Beschränkungen, Auswahl der Einheiten
2. Normierungen: Eingabe, Ausgabe
3. Leistungsfähigkeit: Definierbarkeit von Sprachklassen
4. Hierarchischer Vergleich: Akzeptanz von Sprachen, Berechnung von Funktionen

3. Vorbereitung Blatt 9

3.1 VA 1

Wir betrachten den Beweis zu dem Satz der Vorlesung, in dem eine k-Band-Turingmaschine M durch eine normale Turingmaschine M' simuliert wird.

Geben Sie eine Abschätzung für die Anzahl der Zustände an, die M' haben muss.

Lösung

Wir entwerfen den Zustandsraum von M' wie folgt.

- Zunächst muss natürlich der Zustand der simulierten Turingmaschine gespeichert werden. Sei Q die Zustandsmenge der simulierten Maschine.
- Es werden Zustände benötigt, um die Zeichen zu speichern, die jeweils über den $\star$ standen. Dafür wählen wir $Q_1 = \Gamma^k$.

- Für jede Spur (= jedes simulierte Band) merkt man einen Statuswert, der besagt wie weit man mit der Aktualisierung des Bandes schon gekommen ist. Es gibt vier mögliche Werte:

  U besagt, dass das Zeichen über dem $\star$ noch nicht gelesen wurde.

  R besagt, dass das Zeichen über dem $\star$ schon gelesen wurde.

  W besagt, dass das Zeichen über dem $\star$ gerade aktualisiert wurde. Dabei wurde der $\star$ gelöscht und muss im nächsten Schritt an neuer Position wieder geschrieben werden.

  M besagt, dass der $\star$ wieder geschrieben wurde.

  Damit ist $Q_2 = \{\mathbf{U}, \mathbf{R}, \mathbf{W}, \mathbf{M}\}^k$.

Mit der Zustandsmenge $Q \times Q_1 \times Q_2$ (also insgesamt $|Q| \cdot (4|\Gamma|)^k$ Zuständen) lässt sich nun die Turingmaschine M' beschreiben.

Die Angabe der δ-Funktion erfordert viele Fallunterscheidungen.

Im Detail könnte man natürlich noch optimieren und Zustände einsparen (z.B. sind einige der Zustände nicht erreichbar). Allerdings bleibt in jedem Fall der Term $|\Gamma|^k$ erhalten, da man sich mindestens k Zeichen aus Γ merken muss.

3.2 VA 2

Seien $\Sigma = \{a_1, a_2, \ldots, a_n\}$ ein beliebiges n-elementiges Alphabet und $\Sigma' = \Sigma \cup \{\#\}$.

Geben Sie eine Turingmaschine $M = (Q, \Sigma', \Gamma, \delta, q_0, \square, F)$ mit höchstens 5 Zuständen an, die bei leerer Eingabe das Alphabet Σ in der Form $\#a_1\#a_2 \ldots \#a_n$ auf das Band schreibt und mit dem Kopf auf dem letzten, rechtsstehenden Zeichen der Ausgabe anhält.

Lösung

Sei $M = (\{q_0, q_L, q_A, q_f\}, \Sigma', \Sigma' \cup \{\square\}, \delta, q_0, \square, \{q_f\})$.

Lösung

Sei $M = (\{q_0, q_L, q_A, q_f\}, \Sigma', \Sigma' \cup \{\square\}, \delta, q_0, \square, \{q_f\})$.

Übergang			Bereich	Kommentar
$\delta(q_0, \square)$	$\rightarrow$	$(q_L, \square, L)$		Linkslauf an Wortanfang.
$\delta(q_0, a_i)$	$\rightarrow$	(q_0, a_{i+1}, R)	$i < n$	„Shift“ des Alphabets.
$\delta(q_0, \#)$	$\rightarrow$	$(q_0, \#, R)$		„Shift“ des Alphabets.
$\delta(q_L, a_n)$	$\rightarrow$	(q_f, a_n, N)		Ende.
$\delta(q_L, a_i)$	$\rightarrow$	(q_L, a_i, L)	$i < n$	nach links gehen
$\delta(q_L, \#)$	$\rightarrow$	$(q_L, \#, L)$		nach links gehen
$\delta(q_L, \square)$	$\rightarrow$	(q_A, a_1, L)		Schreiben am Wortanfang.
$\delta(q_A, \square)$	$\rightarrow$	$(q_A, \#, R)$		Schreiben von $\#$.
$\delta(q_A, a_1)$	$\rightarrow$	(q_0, a_1, R)		Zum Start.

3.3 VA 3

Beantworten Sie kurz die folgenden Fragen:

1. Jede unentscheidbare Sprache enthält eine entscheidbare Teilmenge.
2. Jede Teilmenge einer entscheidbaren Sprache ist entscheidbar.
3. Für jede unentscheidbare Sprache A gibt es eine echte Obermenge, die ebenfalls unentscheidbar ist.
4. Aus ,,A entscheidbar“ und ,,$A \cap B$ entscheidbar“ folgt ,,B entscheidbar“.

Eine Menge $A \subseteq \Sigma^*$ heißt entscheidbar, falls die charakteristische Funktion χ_A total auf Σ^* und berechenbar ist.

Dann ergeben sich die folgenden Antworten und Begründungen.

(1) Jede unentscheidbare Sprache enthält eine entscheidbare Teilmenge.

Lösung

Ja, denn jede Sprache $L \subseteq \Sigma^*$ enthält eine endliche Teilmenge, und jede endliche Teilmenge von Σ^* ist entscheidbar.

(2) Jede Teilmenge einer entscheidbaren Sprache ist entscheidbar.

Lösung

Nein, denn $\{0,1\}^*$ ist entscheidbar, aber das allgemeine Halteproblem $H \subseteq \{0,1\}^*$ ist nicht entscheidbar.

(3) Für jede unentscheidbare Sprache A gibt es eine echte Obermenge, die ebenfalls unentscheidbar ist.

Lösung

Ja, denn für ein $w \notin A$ (und das existiert immer, denn Σ^* ist entscheidbar) ist $A \cup \{w\}$ ebenfalls unentscheidbar, falls A unentscheidbar ist.

(4) Aus ,,A entscheidbar“ und ,,$A \cap B$ entscheidbar“ folgt ,,B entscheidbar“.

Lösung

Nein. Gegenbeispiel: $\varnothing$ und $\varnothing \cap H$ sind entscheidbar, weil beide Mengen endlich sind. H (das allgemeine Halteproblem) ist aber nicht entscheidbar.

3.4 VA 4

In einem Tresor liegt eine Liste mit 6-stelligen TAN-Nummern. Der Schlüssel zum Öffnen des Tresors ist verloren gegangen und es gibt keine andere Möglichkeit, den Tresor zu öffnen.

Sei A die Menge der Primzahlen, die auf der TAN-Liste vorkommen.

Ist $A \subseteq \mathbb{N}$ entscheidbar? Begründung!

Lösung

A ist entscheidbar, weil A endlich ist.

Entscheidbarkeit heißt nicht, dass irgend jemand in der Lage sein muss, die Entscheidung zu treffen.

Man muss nur nachweisen, dass es einen Entscheidungsalgorithmus gibt.

Die Begriffe
Berechenbarkeit und Entscheidbarkeit sind Strukturbegriffe.

Bemerkung:

Eine z. B. einelementige Menge $\{a\}$ ist zwar entscheidbar. Dies muss aber nicht bedeuten, dass man dieses eine Element $a \in \Sigma^*$ kennt bzw. zu konstruieren in der Lage ist.

Wir wissen lediglich die abstrakte Existenz dieses a bzw. die Existenz eines Algorithmus, der für vorgelegtes w den Wert der charakteristischen Funktion berechnet, d. h. prüft, ob $w = a$ gilt, wobei der Vergleich $w = a$ für Wörter natürlich mit abbrechendem Algorithmus berechenbar ist.

3.5 VA 5

Man zeige:

1. Sei $\Sigma = \{0, 1\}$ und $f : \Sigma^* \to \Sigma^*$ eine beliebige (möglicherweise partielle) Funktion. Der *Graph* von f ist die Relation $G_f = \{(v, w) \in \Sigma^* \times \Sigma^* ; f(v) = w\}$.
   Wenn G_f entscheidbar ist, dann ist f berechenbar.
2. Gegeben sei eine berechenbare Auflistung (Codierung) aller Turingmaschinen, die jedem Wort $w \in \{0, 1\}^*$ eine Turingmaschine M_w zuordnet. Dann ist die Sprache $L = \{w \mid L(M_w) \text{ ist rekursiv aufzählbar}\}$ entscheidbar.

1. Sei $\Sigma = \{0, 1\}$ und $f : \Sigma^* \to \Sigma^*$ eine beliebige (möglicherweise partielle) Funktion. Der *Graph* von f ist die Relation $G_f = \{(v, w) \in \Sigma^* \times \Sigma^* ; f(v) = w\}$.
   Wenn G_f entscheidbar ist, dann ist f berechenbar.

Lösung

Ausgehend von einem Entscheidungsverfahren für G_f erhält man den folgenden Algorithmus für f:

> *Gegeben ein $v \in \Sigma^*$, zähle alle $w \in \Sigma^*$ nacheinander auf und überprüfe jeweils, ob $(v, w) \in G_f$. Falls ja, dann gib w als Ergebnis zurück. Falls nein, dann fahre mit dem nächsten Wort fort.*

Falls $f(v) \neq \perp$, so wird der Funktionswert irgendwann gefunden. Andernfalls terminiert das Verfahren nicht, was ja genau das geforderte Verhalten ist.

2. Gegeben sei eine berechenbare Auflistung (Codierung) aller Turingmaschinen, die jedem Wort $w \in \{0,1\}^*$ eine Turingmaschine M_w zuordnet. Dann ist die Sprache $L = \{w \mid L(M_w) \text{ ist rekursiv aufzählbar}\}$ entscheidbar.

Lösung

Eine Sprache ist semi-entscheidbar, wenn sie die von einer Turingmaschine akzeptierte Sprache ist.
Definitionsgemäß ist $L(M_w)$ die von M_w akzeptierte Sprache. Also ist $L(M_w)$ semi-entscheidbar für alle w. Damit folgt $L = \{0,1\}^*$, d. h., L ist trivialerweise entscheidbar.