SS 2015

Einführung in die theoretische Informatik

Ernst W. Mayr

Fakultät für Informatik
TU München

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

Sommersemester 2015

Kapitel 0 Organisatorisches

- Vorlesungen:
  - Mo 10:15–12:00, Do 16:00–17:45 MI HS1 F.L. Bauer (MI 00.02.001)
- Übung:
  - 2SWS Tutorübung: siehe Übungswebseite
    Anmeldung in TUMonline
  - 2SWS Zentralübung (nicht verpflichtend): Do 14:30–15:55 MI HS1 F.L. Bauer (MI 00.02.001)
  - Übungsleitung: Dr. Werner Meixner
- Umfang:
  - 4V+2TÜ, 8 ECTS-Punkte
- Sprechstunde:
  - Mo 12:00–13:00 und nach Vereinbarung

- Übungsleitung:
  - Dr. W. Meixner, MI 03.09.040 (meixner@in.tum.de)
    Sprechstunde: Do 12:00–12:30
- Sekretariat:
  - Frau Lissner, MI 03.09.052 (lissner@in.tum.de)

- Übungsaufgaben und Klausur:
  - Ausgabe jeweils am Montag auf der Webseite der Vorlesung
  - Abgabe Montag eine Woche später 13:00Uhr, Kästen bei den Schließfächern im Untergeschoß vor dem HS1
  - Besprechung in der Tutorübung
- Klausur:
  - Endklausur am Donnerstag, 30. Juli 2015, 11:00–14:00, Räume MW2001 (5510.02.001, Rudolf-Diesel-Hörsaal), MI HS1 (MI 00.02.001), Interims-Hörsaal 1 (5620.01.101)
  - Wiederholungsklausur am Donnerstag, 24. September 2015, 08:30–11:30, Raum tba
  - bei den Klausuren sind *keine* Hilfsmittel außer jeweils einem eigenhändig beschriebenen DIN-A4-Blatt zugelassen
  - Für das Bestehen des Moduls ist die erfolgreiche Teilnahme an der Abschlussklausur (mindestens 40% der Gesamtpunktzahl) erforderlich.
  - Die Erfahrungen der letzten Jahre legen nahe, dass es für die erfolgreiche Bearbeitung der Abschlussklausur sehr förderlich ist, die angebotenen Hausaufgabenblätter zu bearbeiten (Sie erhalten sie korrigiert zurück), an der Tutorübung und auch(!) an der (freiwilligen) Zentralübung teilzunehmen!
  - vorauss. 13 Übungsblätter, das letzte am 13. Juli 2015

- Vorkenntnisse:
  - Einführung in die Informatik 1/2
  - Diskrete Strukturen
  - Grundlagen: Algorithmen und Datenstrukturen
- Weiterführende Vorlesungen:
  - Effiziente Algorithmen und Datenstrukturen
  - Automaten, Formale Sprachen, Berechenbarkeit und Entscheidbarkeit
  - Logik
  - Komplexitätstheorie
  - Compilerbau
  - ...
- Webseite:

  http://wwwmayr.in.tum.de/lehre/2015SS/theo/

1. Ziel der Vorlesung

Der Zweck dieser Vorlesung ist das Studium fundamentaler Konzepte in der Theoretischen Informatik. Dies umfasst das Studium der Grundlagen formaler Sprachen und Automaten, von Berechnungsmodellen und Fragen der Entscheidbarkeit, die Diskussion algorithmischer Komplexität sowie einiger grundlegender Konzepte der Komplexitätstheorie.

Themengebiete werden also sein:

- Berechenbarkeitstheorie
  - Betrachtung und Untersuchung der Grenzen, was Rechner überhaupt können
- Komplexitätstheorie
  - Studium der Grenzen, was Rechner mit begrenzten Ressourcen leisten können
  - Herleitung *oberer* und *unterer* Schranken
- Automatentheorie
  - Rechner als endliche Systeme mit endlichem oder unendlichem Speicher
- Grammatiken
  - Aufbau von Programmiersprachen, Ausdruckskraft, Effizienz der Syntaxanalyse
- Algorithmen und ihre Komplexität

Historische Einordnung:

- 1936 Berechenbarkeitstheorie Church, Turing
- 1956 Automatentheorie, Reguläre Ausdrücke Kleene
- 1956 Grammatiken Chomsky
- 1971 Komplexitätstheorie Hennie, Stearns, Cook, Levin

2. Wesentliche Techniken und Konzepte

- Formalisierung und Abstraktion
  - Rechner werden durch mathematische Objekte nachgebildet
  - zu lösende Aufgaben werden mengentheoretisch als Problem definiert
  - die Abfolge von Berechnungsschritten wird formalisiert
  - die quantitative Bestimmung der Komplexität eines Verfahrens bzw. eines Problems wird festgelegt
- Simulation
  - Verfahren zur Ersetzung eines Programms in einem Formalismus A durch ein Programm in einem anderen Formalismus B bei unverändertem Ein-/Ausgabeverhalten

- Reduktion
    - formale Beschreibung für
      "Problem A ist nicht (wesentlich) schwerer als Problem B"
- Äquivalenz
    - ein Formalismus A ist (prinzipiell) genauso mächtig wie Formalismus B
    - "Problem A und Problem B lassen sich (effizient) aufeinander reduzieren"
- Diagonalisierung
    - Auflistung aller Algorithmen einer bestimmten Klasse
    - Beweis durch Widerspruch
    - enger Bezug zu Paradoxa

3. Literatur

Alfred V. Aho, John E. Hopcroft, Jeffrey D. Ullman:
The design and analysis of computer algorithms.
Addison-Wesley Publishing Company, Reading (MA), 1976

John E. Hopcroft, Jeffrey D. Ullman:
Introduction to automata theory, languages, and computation.
Addison-Wesley Publishing Company, Reading (MA), 1979

Uwe Schöning:
Theoretische Informatik — kurzgefasst.
Spektrum Akademischer Verlag GmbH, Heidelberg-Berlin, 1997

Katrin Erk, Lutz Priese:
Theoretische Informatik: Eine umfassende Einführung (3. Auflage).
Springer-Verlag, Berlin-Heidelberg-New York, 2008
http://link.springer.com.eaccess.ub.tum.de/book/10.1007%2F978-3-540-76320-8

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest:
Introduction to algorithms.
McGraw-Hill Book Company, New York-St. Louis-San Francisco-Montreal-Toronto, 1990

Thomas Ottmann, Peter Widmayer:
Algorithmen und Datenstrukturen.
B.I., Mannheim-Leipzig-Wien-Zürich, 1993
http://link.springer.com.eaccess.ub.tum.de/book/10.1007%2F978-3-8274-2804-2

Volker Heun:
Grundlegende Algorithmen.
Vieweg, 2000
http://link.springer.com.eaccess.ub.tum.de/book/10.1007%2F978-3-322-80323-8

Ingo Wegener:
Theoretische Informatik.
B.G. Teubner, Stuttgart, 1993
http://link.springer.com.eaccess.ub.tum.de/book/10.1007%2F978-3-322-94004-9

Kapitel I Formale Sprachen und Automaten

1. Beispiele

Sei Σ ein (endliches) Alphabet. Dann

Definition 1

1. ist ein Wort/String über Σ eine endliche Folge von Zeichen aus Σ;
2. wird das leere Wort mit ϵ bezeichnet;
3. bezeichnet uv die Konkatenation der beiden Wörter u und v;
4. ist Σ^* das Monoid über Σ, d.h. die Menge aller endlichen Wörter über Σ;
5. ist Σ^+ die Menge aller nichtleeren endlichen Wörter über Σ;
6. bezeichnet $|w|$ für $w \in \Sigma^*$ die Länge von w;
7. ist, für jedes Wort w, w^n definiert durch $w^0 = \epsilon$ und $w^{n+1} = ww^n$;
8. ist Σ^n für $n \in \mathbb{N}_0$ die Menge aller Wörter der Länge n in Σ^*;
9. eine Teilmenge $L \subseteq \Sigma^*$ eine formale Sprache.

Beispiel 2

Wir betrachten folgende Grammatik:

$$\begin{array}{rcl}
\langle\text{Satz}\rangle & \rightarrow & \langle\text{Subjekt}\rangle\langle\text{Prädikat}\rangle\langle\text{Objekt}\rangle \\
\langle\text{Subjekt}\rangle & \rightarrow & \langle\text{Artikel}\rangle\langle\text{Attribut}\rangle\langle\text{Substantiv}\rangle \\
\langle\text{Artikel}\rangle & \rightarrow & \epsilon \\
\langle\text{Artikel}\rangle & \rightarrow & \text{der}|\text{die}|\text{das}|\text{ein}|\dots \\
\langle\text{Attribut}\rangle & \rightarrow & \epsilon|\langle\text{Adjektiv}\rangle|\langle\text{Adjektiv}\rangle\langle\text{Attribut}\rangle \\
\langle\text{Adjektiv}\rangle & \rightarrow & \text{gross}|\text{klein}|\text{schön}|\dots
\end{array}$$

Die vorletzte Ersetzungsregel ist rekursiv, die durch diese Grammatik definierte Sprache deshalb unendlich.

Beispiel 3 (Formale Sprachen)

- die Menge aller Wörter in der 24. Auflage des Duden
- $L_1 = \{aa, aaaa, aaaaaa, \dots\} = \{(aa)^n; n \in \mathbb{N}\} \qquad (\Sigma_1 = \{a\})$
- $L_2 = \{ab, abab, ababab, \dots\} = \{(ab)^n; n \in \mathbb{N}\} \qquad (\Sigma_2 = \{a, b\})$
- $L_3 = \{ab, aabb, aaabbb, \dots\} = \{a^n b^n; n \in \mathbb{N}\} \qquad (\Sigma_3 = \{a, b\})$
- $L_4 = \{a, b, aa, ab, bb, aaa, aab, abb, bbb \dots\}$
  $= \{a^m b^n; m, n \in \mathbb{N}_0, m + n > 0\} \qquad (\Sigma_4 = \{a, b\})$
- $L_5 = \emptyset$
- $L_6 = \{\epsilon\}$
- $L_7 = \{x \in \Sigma^*;$ ein gegebenes Programm/TM hält bei Eingabe $x\}$

Beispiel (Forts.)

- *Die "Menge" der Sätze in deutscher Sprache ist keine formale Sprache*
- ϵ *ist keine formale Sprache*
- $\mathbb{R}$ *ist keine formale Sprache*