Automaten, Formale Sprachen (und Berechenbarkeit) (WS 2006/07)

Markus Holzer

Institut für Informatik, Technische Universität München,
Boltzmannstraße 3, D-85748 Garching bei München, Germany
email: holzer@in.tum.de

Allgemeines

Bereich: 4+2 SWS im Bereich Informatik III

Inhaltsangabe und Literaturverzeichnis

1. Automaten und formale Sprachen
   (a) Allgemeines zu Automaten und formale Sprachen, Berechenbarkeit, einige Beispiele.
   (b) Grundlagen: Alphabet, Wort, leeres Wort, Konkatenation, Monoid, freies Monoid, Kleenesche Hülle.
   (c) Endliche Automaten: Definition deterministischer (DFA) und nichtdeterministischer (NFA) endlicher Automat, Potenzautomatenkonstruktion, Myhill-nerode Äquivalenzrelation, Satz von Myhill-Nerode, Äquivalenzrelation über Automaten, Beziehung der beiden Äquivalenzrelationen, Minimalautomat, Eigenschaften des Minimalautomaten bzgl. der Myhill-Nerode Äquivalenzrelation, Zustandexplosion bei Umwandlung NFA nach DFA, Minimierungsalgorithmus und Beispiel.
   (d) Alternative Darstellung regulärer Mengen:
      i. Syntaxdiagramme oder beschriftete Myhill-Graphen: Definition lokale Menge, Definition Myhill-Graph, Wegemengen, jede lokale Menge ist reguläre aber nicht umgekehrt, jede reguläre Menge ist homomorphes Bild einer lokalen Menge,
      ii. Reguläre Ausdrucke: Definition regulärer Ausdruck, Äquivalenz von regulären Ausdrücken und NFA's (Richtung NFA nach regulären Ausdruck mit Hilfe eines graphischen Verfahrens).
      iii. Erkennbarkeit: Definition von Erkennbarkeit, Beziehung zu endlichen Monoiden, Transitionsmonoid des DFA's, syntaktische Relation und Monoid, Erkennbarkeit und syntaktisches Monoid, Beziehung syntaktisches Monoid und Transitionsmonoid.

iv. Rechnen mit regulären Ausdrücken und Gleichungssystem: Leerworteigenschaft, Axiomensystem nach Salomaa, Rechenregeln (Kongruenz, Symmetrie, Transitivität und Gleichungsauflösung), Gleichungssystem eines endlichen Automaten, Matrizendarstellung, Bestimmung der Lösung durch Iteration.
v. endliche Zweiwege Automaten: Definition deterministischer Automat, Äquivalenz zu normalen endlichen Automaten mittels Crossing-Sequence Argument.
vi. Monadische Logik 2. Stufe: Definition von FO und MSO, Definition von FO- und MSO-Erkennbarkeit, einfache Beispiele, Einbettung von FO in MSO, Äquivalenz von regulären Mengen und MSO-Erkennbarkeit per struktureller Induktion.

(e) Endliche Automaten über unendlichen Wörtern: Grundlagen zu unendlichen Wörtern, Mengen, Definition ω-regulären Mengen, Darstellung als endliche Vereinigung.
i. Büchi Automaten (BA): Definition, periodische Wörter, Büchi akzeptierbar gleich ω-regulär, Verhältnis Determinismus und Nichtdeterminismus, Pfeiloperation, Beispiele zur Pfeiloperation, DBA's und die Pfeiloperation, DBA Sprachen sind iechte Teilmenge der ω-regulären Sprachen, Abschlußeigenschaften von DBA Sprachen (nicht unter Komplementbildung, Abschluß unter Schnitt und Vereinigung).
ii. Muller Automaten (MA): Definition, (D)MA Sprachen sind ω-regulär, Aschlußeigenschaften von DMA Sprachen (Vereinigung, Schnitt, Komplementbildung), Darstellung von DMA Sprachen als endliche Vereinigung von Differenzen von DBA Sprachen.
iii. Rabin Automaten (RA): Definition, Äquivalenz von DRA Sprachen und DMA Sprachen, BA Sprachen sind auch DRA Sprachen, hierzu verallgemeinerte Potenzautomatenkonstruktion nach Safra, Speicherung der Mengen in Bäumen, Endlichkeit dieser Bäume, großes Beispiel, Beweisskizze für die Korrektheit, Zustandsexplosion, BA Sprachen sind unter Komplementbildung abgeschlossen.

(f) Abschließende Übersicht.

2. Berechenbarkeit
(a) Allgemeines zu Automaten und formale Sprachen, Berechenbarkeit, einige Beispiele.
(b) Grundlagen: Mengen, Relationen, Bild und Urbild, Abbildungen (injektiv, surjektiv, bijektiv), Kardinalitätsgleichheit von Mengen, Abzählbarkeit und Überabzählbarkeit, Satz von Schröder-Bernstein, Kardinalitätsargumente (nat. Zahlen, reelle Zahlen, offenes Einheitsintervall), Satz von Cantor, Diagonalisierung.
(c) While-Programme: Syntax, Sematik, Marko-Anweisungen, Definition von effektiv berechenbar, Church-Turing These, andere Möglichkeiten der Formalisierung.
(d) Standardtechniken: Gödelsierung, Standardaufzählung von Funktionen, nicht Berechenbarkeit des Halteproblems, universelle Funktion, Enumerationstheorem, s-m-n Theorem.

(e) Unentscheidbare Probleme: Definition von Entscheidbarkeit, Halteproblem, Totalitätsproblem—es gibt totale Funktionen die nicht berechenbar sind, allgemeines Halteproblem, Identitätsproblem, Äquivalenzproblem.

(f) Rekursionstheorem und Eigenschaften von Aufzählungen: Rekursionstheorem, Beispiele mit einfachen Programmtransformationsfunktionen, Anzahl der Fixpunkte, Selbstreferenzierung und Programme die ihren eigenen Quellcode ausgeben, akzeptable Programmiersysteme, Rogers Isomorphie Theorem für akzeptable Programmiersysteme.

(g) Berechnbare Eigenschaften von Mengen: Definition Entscheidbarkeit, einfache Eigenschaften von entscheidbaren Mengen, Definition rekursive Aufzählbarkeit, Beziehungen von entscheidbare und rekursiv aufzählbaren Mengen, einfache Beispiele entscheidbarer und rekursiv aufzählbarer Mengen, Entscheidbarkeit entspricht rekursiver Aufzählbarkeit und co-Aufzählbarkeit, Entscheidbarkeit entspricht rekursiver Aufzählbakeit in aufsteigender Reihenfolge, alternative Charakterisierungen von rekursiver Aufzählbarkeit durch Bild- und Urbildbereich von Funktionen, Standardaufzählung von rekursiv aufzählbaren Mengen, einfache Operationen auf rekursiv aufzählbaren Mengen.

(h) Die Sätze von Rice: Definition Indexmenge die Funktionen respektiert, Satz von Rice, einfache Beispiele, Satz von Rice—Variante I und II bzgl. rekursiver Aufzählbarkeit, Definition many-one Reduktion, einfache Eigenschaften von many-one Reduktionen, Definition Turing Reduktion, einfache Eigenschaften von Turing Reduktionen, Vollständigkeit der Menge K (Menge der Indizes so, daß i auf Programm i terminiert).