Gedächtnisprotokoll Vertiefungsprüfung
Automaten, Logik und Verifikation

Professor Wolfgang Thomas
Lehrstuhl 7 für Informatik

Stefan Buhr
Email: stefan.buhr@rwth-aachen.de

4. Oktober 2004

Zusammenfassung

Dieses Prüfungsprotokoll versucht einige Themen zusammenzutragen, die im Laufe meiner Prüfung von Professor Thomas angeschnitten wurden. Leider kriege ich die Fragen nicht mehr 100%ig zusammen, noch weniger *meine* Antworten. Insofern kann ich keine Garantie auf Vollständigkeit geben. Auch kann ich nicht immer angeben, worauf Professor Thomas hinauswollte. Hier stichpunktartig eine kurze Übersicht über die Prüfung:
Datum der Prüfung: 4. Oktober 2004, 10.00 Uhr
Prüfer: Professor Wolfgang Thomas, Lehrstuhl 7 für Informatik
Beisitzer: Philipp Rohde, Lehrstuhl 7 für Informatik
Geprüfte Fächer:

- Angewandte Automatentheorie (SS 2003)
- Automata and Reactive Systems (WS 2002/03)
- Model-Checking (WS 2003/04)
- Theorien/Spiele/Algorithmen (Prof. Grädel, WS 2003/04)

Dauer: ca. 50 Minuten
Prüfungsnote: 1.3

1 Angewandte Automatentheorie

Baumautomaten

- Wie funktionieren Baumautomaten? Wie findet der Einstieg in einen Baum statt? Gibt es beim Vergleich von Baum- mit Wortautomaten Unterschiede?

  Auch in meiner Prüfung die Erläuterung, dass deterministische Top-Down-Automaten echt schwächer sind als nichtdeterministische. Ich musste jedoch kein Beispiel angeben.

- Mit Hilfe welcher Verknüpfungen werden reguläre Ausdrücke über Bäumen gebildet? Was sind spezielle Bäume?

  Ich habe die Verknüpfungen aufgezählt und spezielle Bäume erläutert.

- Unter welchen Operationen sind denn deterministische Baumautomaten abgeschlossen? *Produkt (Schnitt), Vereinigung, Komplement. Vorgehensweise und Komplexität der einzelnen Schritte erläutert.*

- Kann man algorithmisch testen, ob ein Baumautomat überhaupt irgendeinen Baum erkennt?

  Ja, man kann die Leerheit algorithmisch testen, und zwar über eine Art Potenzmengenkonstruktion. Man notiert an den Blättern alle Zustände, die einen Einstieg in den Baum erlauben, wendet erlaubte Transitionen an und prüft, ob an der Wurzel ein Startzustand auftritt.

Petrinetze

- Was sind Petrinetze? Was ist ein Zustand eines Petrinetzes?

  Erläuterungen zur Funktionsweise von Petrinetzen, Markierungen.

- Welche Probleme kennen Sie in Bezug auf Petrinetze?

  Erreichbarkeitsproblem, Überdeckungsproblem, Deadlock-Freiheit, Beschränktheitsproblem

- Was ist das Überdeckungsproblem?

Definition

- Wie testet man das Erreichbarkeitsproblem?

  Ich weiss noch nicht genau, was hier die ideale Antwort ist. Aber irgendwie reicht der Karp-Miller-Baum hier in seiner groben Sicht wohl nicht so ganz aus.

- Welche Sprachen können Petrinetze erkennen?

  Inklusionen grafisch dargestellt.

- Welche Sprache ist denn kontextfrei, und ist Petrinetz-erkennbar? Gibt es eine kontextfreie Sprache, die nicht Petrinetz-erkennbar ist?

  $a^n c b^n$ *erkennbar,* ww^R *(Menge der geraden Palindrome) nicht. Auf Nachfrage erläutert, warum.*

2 Automata and Reactive Systems

- Deterministische Büchi-Automaten: sind die genauso mächtig wie das nichtdeterministisch Gegenstück?

  Nein, sie sind schwächer. Erkennen z. B. $(0+1)^*0^\omega$ *nicht. Auf Nachfrage bewiesen, warum.*

- Gibt es deterministische Automaten, die genauso mächtig sind wie nichtdeterministische Büchi-Automaten?

  Ja, deterministische Muller-Automaten.

- Kann man prüfen, ob eine Sprache bereits deterministisch Büchi-erkennbar ist? *Ja, prüfe auf Eigenschaft „closed under superloops“.*

- Was genau heißt das? *Genau erläutert: jeder Loops, der Obermenge bzgl. Mengeninklusion eines akzeptierenden Loops ist, ist wieder akzeptierend.*

- Welches ist die einfache Richtung im Beweis? Von Muller und „closed under superloops“ zu deterministisch Büchi oder umgekehrt?

Muller und „closed under superloops“ zu deterministisch Büchi ist recht einfach. Erläutert.

- Wofür braucht man Paritätsspiele? Wie sehen sie aus?

  Erläutert, wer gegen wen um was spielt. Komplementierung von Paritätsautomaten. Vorteil: Paritätsspiele sind determiniert, deshalb kann aus der Nichtexistenz einer Gewinnbedingung für den Einen die Existenz einer Gewinnbedingung für den Anderen gefolgert werden.

3 Model-Checking

- Wie funktioniert LTL-Model-Checking? Welche Komplexität haben die einzelnen Schritte?

  Komplexität im Detail erläutert, damit die Frage vorweggenommen, welcher Schritt zum exponentiellen „Blow-Up“ führt.

- Wie kann man denn in LTL „irgendwann wird ein bestimmter Zustand erreicht“ in LTL ausdrücken?

  Zustand bekommt Label p und dann durch $EF\,p$.

- Was ist mit dem CTL-Model-Checking? (Irgendwie hat den Bogen zu *Symbolischem MC* gekriegt!)

  Auf Nachfrage recht detailliert den Aufbau von OBDDs, die Kodierung von Kripke-Struktur und Formel (bzw. Formel-Komplement) als OBDD dargestellt.

- Gibt es einen Äquivalenztest für OBDDs? Wie könnte der funktionieren?

  OBDDs sind effizient minimierbar. Nach Reduktion Test auf Isomorphie in linearer Zeit bzgl. Kanten- und Knotenzahl.

- Wie funktioniert das „normale“ Model-Checking für CTL?

  Erläutert, mit welchen Formeln man auskommt und dann kurz erläutert, wie's funktioniert.

4 Theorien, Spiele, Algorithmen

Direkt vorweg: meine *absolute* Schwachstelle. Komplizierte Vorlesung, hab' wenig verstanden.

- Wie sind Formeln der Logik L_μ aufgebaut? Was bedeutet das μ bzw. ν in $\mu X.\varphi(X)$ bzw. $\nu X.\varphi(X)$?
- Was sind die Model-Checking-Spiele für L_μ? Wie funktioniert das?

  Spielpositionen erläutert, Entfaltung von Paritätsspielen angeschnitten.
- Komplexität des Model-Checking für L_μ?

Fazit

Prüfungsatmosphäre erwartungsgemäß locker. Der Beisitzer hat im Prinzip kein Wort gesagt, und somit weder zu meiner Beruhigung noch zu meiner Nervosität beigetragen.