

Prüfungsprotokoll

Alexander Wilden

E-Mail: october dot rust at gmx dot de

25.4.2006

Allgemeines

- Prüfer / Themen:
 - Thomas:
 - * Automata on infinite words (WS05/06)
 - * Unendliche Spiele (WS05/06)
 - * Compilerbau (SS05)
 - Katoen:
 - * Model Checking (WS05/06)
- Note: 1,7

Hinweis

Da ich nach der Prüfung schon angefangen hatte zu feiern, bevor ich mir Notizen zu den Inhalten gemacht habe, ist dieses Protokoll eventuell etwas ungenau und bruchstückhaft. Ich hatte aber einem Komillitonen versprochen ein Protokoll zu schreiben, deswegen versuche ich das hiermit :-). Ich hoffe es hilft trotzdem jemandem. Ansonsten darf man mich gerne zu einzelnen Sachen persönlich befragen.

1 Automata on infinite words

Er: Aufbau einer ω -regulären Sprache.

Ich: $L = \bigcup_{i=1}^n U_i \cdot V_i^\omega$ mit U_i, V_i regulär

Er: Woher kommt das n bei der Vereinigung

Ich: Anzahl der Endzustände des Büchautomaten

Er: Frage nach der Darstellung der U_i, V_i

Ich: Hatte nicht ganz verstanden was er meinte, aber es ging wahrscheinlich um die Darstellung als $W_{q_0, q}$ und $W_{q, q}$

Da ich hier auf dieses $\sim_{\mathcal{A}}$ zu sprechen gekommen bin, wollte er jetzt wissen, was es damit auf sich hatte, habe es aber irgendwie nicht mehr hinbekommen das zu erklären. Er ist aber zum Glück auch nicht weiter drauf eingegangen. War wohl mein Fehler, das überhaupt zu erwähnen ;-)

Er: Unterschied det und nicht-det. Büchi

Ich: det. Büchi ist schwächer, da er die Sprache mit nur endlich vielen b $L = (a+b)^* a^\omega$ nicht erkennt. Habe hier ausserdem erwähnt, dass man durch die Muller-Schupp Konstruktion einen det. Automaten aus dem nicht-det. Büchi Automaten erzeugen kann.

Er: Aufbau eines Rabin Automaten

Ich: Erklärt und kurz skizziert, wie die Akzeptanzpaare aussehen und was diese bedeuten

Er: Die MS-Trees haben ja eine bestimmte Eigenschaft ...

Ich: Sie sind strikt binär

Er: Frage nach der Bezeichnung der Knoten eines MS-Trees und dem Aufbau MS-Trees.

Ich: Knotennamen aus Q , neue Knotennamen aus Namensreservoir $3 * |Q|$, Färbung (er wollte gar nichts genauer zu den Farben wissen), Anordnung der Knoten (final links, non-final rechts)

Er: Frage nach der Größe des Zustandsraums

Ich: $2^{O(n \log n)}$

Er: Gehts auch mit weniger

Ich: Erwähne das Theorem von Löding, in dem bewiesen wurde, dass es nicht besser geht, wegen der Automatenklasse, die $n!$ Zustände benötigt und damit auch in $2^{O(n \log n)}$ liegt.

Er: Wir haben ja auch eine Logik kennengelernt

Ich: S1S und wir haben gezeigt, dass eine ω -reguläre Sprache genau dann S1S-

definierbar ist, wenn sie Büchi-erkennbar ist.

Er: Dann beweisen Sie mal eine Richtung. Suchen Sie sich die aus, die Ihnen am besten liegt.

Ich: Habe angefangen die Umwandlung von Büchi zu S1S hinzuschreiben, er hat mich aber unterbrochen und nur gefragt was für Elemente in die Formel einfließen (also Startzustand, Zustandsübergänge und unendlich oft ein Endzustand).

Er: Wir haben ja noch eine andere Logik kennengelernt.

Ich: LTL

Er: Ja, genau. Was können Sie mir denn zu der Beziehung zwischen LTL und S1S sagen?

Ich: LTL ist schwächer, da wir jede Formel in einen Büchi-Automaten umwandeln können, aber nicht umgekehrt. Habe dann noch die non-counting property angesprochen und die Sprache $L = (00)^*1^\omega$ erklärt.

Er: Kann man denn sowas wie 001001001001... LTL erkennen.

Ich: Habe zunächst etwas gegrübelt, dann aber sowas wie $0 \wedge X0 \wedge XX1$ aufgeschrieben. Er war damit auch halbwegs zufrieden und meinte nur, dass das halt immer gelten müsste, also ein \forall davor müsste, aber der Ansatz wäre richtig.

2 Unendliche Spiele

Er: Nennen Sie mal ein paar Arten von Spielen

Ich: Paritätsspiele, schwache Paritätsspiele, Muller-Spiele, ...

Er: *lacht*, fangen Sie doch mal mit was einfacherem an.

Ich: Erreichbarkeitsspiele.

Er: Ok, erklären Sie mal wie ein Staiger-Wagner Spiel aussieht.

Ich: erklärt.

Er: Nennen Sie mal die Strategie für Erreichbarkeits- und Büchi-Spiele.

Ich: Angefangen die Attraktormengen aufzuschreiben. Hat mich da unter-

brochen, da er wohl gemerkt hat, dass ich die kannte. Die Recurrencemengen habe ich auch nur so gerade angefangen aufzuschreiben und habe mehr erklärt, was diese bedeuten (also Endzustände, von denen aus wir eine Rückkehr in selbige garantieren können).

Er: Was ist denn der Hauptunterschied bei der Definition der Attraktormengen und Recurrencemengen.

Ich: Bei den Attraktormengen haben wir eine Vereinigung und bei den Recurrencemengen einen Schnitt mit den Endzuständen.

Er: Kann man diese Strategien bei einem unendlichen Spielgraphen anwenden?

Ich: War da zunächst etwas verwirrt und meinte dann, dass das wohl eher nicht ginge. Er sagte dann, theoretisch könne man das schon berechnen, aber es wäre nicht sehr effektiv, da die Attraktormengen ja im allgemeinen dann unendlich groß werden.

Er: Zeigen Sie mir mit einem Spiel ihrer Wahl, dass nicht immer eine positionale Strategie reicht.

Ich: habe da VE-Spiele genommen und ein Beispiel aus der Vorlesung, dass eigentlich für Muller-Spiele sein sollte, aber auch hier passte.

Er: Wie können wir das denn lösen.

Ich: Mit einem Speicher.

Er: Wie groß muss denn der Speicher sein?

Ich: Endlich, man muss sich nur die schon besuchten Zustände merken.

Er: Ein Bit täts hier schon (galt aber nur für das Beispiel, das ich genommen hatte). Wie kann man denn allgemein sowas lösen

Ich: Mit Strategieautomaten und per Reduktion auf Spiele, für die wir eine positionale GWS haben.

Er: Was können Sie zu Baumautomaten sagen?

Ich: Büchi-Baumautomaten sind nicht unter komplement abgeschlossen, Paritätsbaumautomaten aber wohl.

Er: Wie kann man denn den Abschluss unter Komplement bei PBAs zeigen?

Ich: Übersetzung in Spiel mit Automat und Pfadsucher.

Er: meinte dann das wäre ja auch problematisch da zu bestimmen, ob der Lauf akzeptierend ist oder nicht, da die Bäume ja unendlich sind. Er wollte an dieser Stelle auf reguläre Bäume hinaus, hatte es auch auf der Zunge, habs aber dummer Weise nicht gesagt.

3 Compilerbau

(Habe das hier mal wirklich nur in Stichworten aufgeschrieben, da es schon einige Protokolle zu CP gibt und das ja auch eigentlich nicht so schwer ist.)

- was die lexikalische Analyse ist
- was first-longest-match bedeutet und wofür man es braucht
- wie wir das machen (also reguläre Ausdrücke und Thompsonkonstruktion usw., wollte da wohl noch hören, dass man die regulären Ausdrücke auf "Disjunktheit" prüfen sollte, also dass Bezeichner nicht in die selben Ausdrücke wie Anweisungen fallen können usw.)
- was Syntaxanalyse ist
- Frage nach Algorithmus aus Grundstudium für diese Aufgabe (CYK)
- Frage wieso wir CYK nicht benutzen, wäre doch toll son Algorithmus der alle CFG erkennt :-) (habe gesagt, dass der halt quadratischen Platzbedarf und kubischen Zeitaufwand hat und dass wir das hier nicht brauchen, da wir spezielle Grammatiken betrachten)
- Frage nach der Definition von $LL(k)$, la -Mengen und die zugehörigen Hilfsmittel (also follow und first Mengen)
- Frage nach Kellerinhalten des TD-Automaten
- Zu $LR(0)$ und $LR(1)$: Kellerinhalte des BU-Automaten
- welche Funktionen es dort gibt (action und goto Funktion erklärt)
- kurze Frage wie in der semantischen Analyse bei Attributgrammatiken denn Zirkularitäten entstehen können (sollte Beispiel aufzeichnen, wichtige Bemerkung war wohl, dass man den "Deckel" finden muss, damit die Abhängigkeit komplett ist)

4 Model Checking

(Hier der Teil ist leider auch nur in Stichworten, da es manchmal recht schwer war zu verstehen, worauf Herr Katoen hinauswollte. Deswegen ist er bei weitem kein schlechter Prüfer, aber eventuell sollte man nachfragen, ob man den Teil nicht in Englisch machen könnte, wenn man selber Englisch auch gut beherrscht. Das wäre wahrscheinlich für beide Seiten etwas angenehmer gewesen.)

- was für Arten von Fairness es gibt (uncond, strong, weak)
- Synchronprodukt von NBA / NFA und TS und ich sollte sagen was das Ergebnis davon ist (ein Transitionsystem)
- was eine Persistence-Property ist
- wie wir das checken
- wie der nested depth first search algorithmus funktioniert (hier bin ich arg ins Stocken geraten, da ich mir die ganzen Algorithmen nicht besonders gut und im Detail angeschaut hatte)
- Fairness in LTL, wie wir die da Überprüfen können
- LTL-Model Checking (also NBA zu LTL Formel, dann Synchronprodukt, dann nested depth first search)
- Fairsatisfaction bei CTL (hier hat er mich auf dem total falschen Fuß erwischt ... hätte mir das wohl was besser anschauen sollen :-)
- Dann wollte er was wissen mit CTL* und LTL. Ich habe nicht ganz verstanden, was er da von mir hören wollte, hatte mir aber irgendwie schon fast die Einbettung von LTL in CTL* gedacht, was es auch war. Habs noch gerade so hingekritzelt bekommen ($\forall\phi \dots$) und er hat da wohl gemerkt, dass ich es wusste, aber dass ich einfach nicht drauf gekommen bin, dass er das von mir sehen wollte :-)
- Erklären was Bisimilar und Simulation Order bedeutet und was daraus folgt
- Frage nach Properties die nicht liveness und nicht safety sind (ging darum dass diese dann eine Konjunktion von Liveness und safety darstellen, habs auch gerade noch so sagen können: $P = P_{safe} \cap P_{live}$)