

# Gedächtnisprotokoll

## Diplomprüfung Theoretische Informatik

Klaus Meyer

6. November 2006

### Allgemeines

- **Vorlesungen:**
  - Automata on Infinite Words (Thomas, WS 2005/06)
  - Unendliche Spiele (Thomas, WS 2005/06)
  - Infinite State System Verification (Thomas, SS 2006)
  - Rekursionstheorie (Thomas, SS 2006)
  - Compilerbau (Indermark, SS 2005)
- **Prüfer:** Prof. Dr. Dr.h.c. Wolfgang Thomas
- **Beisitzer:** Dipl.-Inform. Jan Altenbernd
- **Datum:** 06.11.2006
- **Dauer:** ca. 50 Minuten
- **Vorbereitungszeit:** 5 1/2 Wochen (sehr knapp!)

Ich habe mich bemüht, dieses Protokoll möglichst genau und detailliert zu halten. Das könnte interessant sein, gerade weil meine Prüfung teilweise etwas holprig verlaufen ist. Auf einen einfachen Fragenkatalog kann man sie daher eh nicht reduzieren.

Ich kann natürlich keine Garantie auf Vollständigkeit geben (der übliche Disclaimer), aber wenn etwas fehlt, dann nicht viel.

## 1 Automata on Infinite Words

- **Wir hatten verschiedene Automaten kennengelernt: Büchi, co-Büchi, Muller... Wie sind denn co-Büchi-Automaten definiert?**

Es gibt die Akzeptanzkomponente  $F$ , eine Menge von Zuständen, und der Automat akzeptiert genau dann, wenn ab einem bestimmten Punkt alle Zustände in  $F$  sind.

- **Ganz genau. Nun gibt es Büchi- und co-Büchi-Automaten, deterministische und nichtdeterministische. Gibt es da Unterschiede?**

co-BA sind deterministisch, und DBA sind schwächer als NBA.

**Also ist die Menge der DBA-Sprachen in der Menge der NBA-Sprachen enthalten. Und wie sind co-BA da enthalten? Gibt es da einen Zusammenhang mit BA?**

co-BA erkennen gerade die Komplemente der BA-erkennbaren Sprachen.

- **Also sind det. BA und co-BA unterschiedlich in NBA enthalten. Nennen Sie mal eine Sprache, die Büchi-erkennbar ist.**

Nichtdeterministisch?

**Ach so, ich habe mich unklar ausgedrückt. Eine Sprache, die NBA-, aber nicht DBA-erkennbar ist.**

$(0^*1)^\omega$

**Hm, wahrscheinlich verwechseln Sie da etwas. Ich behaupte, dass diese Sprache DBA-erkennbar ist.**

Ach richtig, der Automat kann immer akzeptieren wenn eine 1 gesehen wird. . . Dann besser die Sprache, in der irgendwann nur noch Einsen vorkommen.

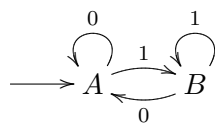
- **Ist die Sprache, die Sie aufgeschrieben haben, co-Büchi-erkennbar?**

Nein.

- **Gibt es denn deterministische Automaten, die alles das erkennen können?**

Muller-Automaten.

- **Dann möchte ich gerne etwas über die Beziehung zwischen Büchi- und Muller-Automaten wissen. Geben Sie mal einen Muller-Automaten für Ihre Sprache an.**



- **Wie sieht denn die Akzeptanzkomponente eines Muller-Automaten aus?**

$\mathcal{F}$  ist eine Teilmenge der Potenzmenge von  $Q$ . Für diesen Automaten ist  $\mathcal{F} = \{\{B\}, \{A, B\}\}$ .

- **Wir hatten ja den Satz von Landweber, was sagt der denn hierüber aus?**

Eine Muller-erkennbare Sprache ist genau dann det.-Büchi-erkennbar, wenn  $\mathcal{F}$  unter Superloops abgeschlossen ist.

- **Ist dieses  $\mathcal{F}$  unter Superloops abgeschlossen?**

Ja,  $\{A, B\}$  ist die einzige Superloop von  $\{B\}$ .

- **Ist es auch unter Subloops abgeschlossen?**

Nein,  $\{A\}$  ist Subloop von  $\{A, B\}$ .

**Genau, das zeigt übrigens dass der Satz stimmt, aber das ist natürlich kein Beweis.**

- **Es muss ja möglich sein, DBA in MA umzuwandeln. Wie macht man das?**

Hm, man könnte alle erreichten Zustände aufsammeln. . . *\*überleg\**

**Also, auf der einen Seite haben Sie ein  $F$ ...**

Ach natürlich, die Zustände braucht man gar nicht ändern, sondern setzt  $\mathcal{F}$  auf die Mengen von Zuständen, die mit  $F$  einen nichtleeren Schnitt bilden.

- **Ja, es reichen auch nur die Schleifen davon. So, wie wandelt man NBA in MA um?**

Das geht mit der Muller-Schupp-Konstruktion.

- **Es gibt da noch eine andere Konstruktion, kennen Sie den Namen?**

Safra.

- **Erklären Sie doch mal in groben Zügen, wie die Muller-Schupp-Konstruktion funktioniert.**

Man konstruiert einen Rabin-Automaten, dessen Zustände die Muller-Schupp-Trees von  $Q$  sind.<sup>1</sup>

---

<sup>1</sup>Zum Glück habe ich am Abend vor der Prüfung dies gefunden:  
<http://www-i7.informatik.rwth-aachen.de/download/papers/wallmeier/stw05.pdf>  
 Siehe auch <http://www-i7.informatik.rwth-aachen.de/d/research/omegadet.html>

- **Ach richtig, wir haben das für Rabin-Automaten gezeigt. Können Sie etwas mehr über die MS-Trees erzählen?**

Sie sind strikt binär, und sind beschriftet mit natürlichen Zahlen, wobei endlich viele reichen, nämlich...

- **Womit sind die MS-Trees denn noch beschriftet?**

Teilmengen von  $Q$ , und Farben rot, grün und gelb.

- **Was ist denn die Akzeptanzbedingung?**

Es müssen unendlich oft MS-Trees mit grünen Knoten erreicht werden, und nur endlich oft Bäume die... *\*grübel\**

**Ich schlage mal eine Bedingung vor, und Sie sagen, ob das so richtig ist: Unendlich oft Bäume, in denen die Farbe grün vorkommt.**

Das ist nötig, aber das reicht nicht.

**Was fehlt denn noch?**

Irgendwann in dem Lauf gibt es nur noch einen Baum, äh, Knoten, der sich nicht mehr ändert...

- **Richtig, das soll mal reichen. Die Frage nach der Komplexität lassen wir mal... Wieviele Zustände erhält man, wenn der Büchi-Automat  $n$  Zustände hat?**

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

- **Geht das denn noch besser?**

Nein.

- **Für welche Automaten gilt das? Ich meine, diese Komplexität gilt ja nur für Rabin-Automaten. Wie sieht es bei Muller-Automaten aus? Es kann ja sein, dass man da durch geschicktes Ausnutzen von Schleifen etwas machen könnte...**

Keine Ahnung.

Ich habe in der Vorlesung mal gesagt, dass das noch offen ist, aber das ist inzwischen gelöst.

## 2 Unendliche Spiele

- **Wir hatten verschiedene Spiele und Strategien betrachtet. Was für Strategien kennen Sie denn?**

Die Attraktorstrategie für Erreichbarkeitsspiele...

- **Diese Strategie hat eine bestimmte Eigenschaft.**  
Sie ist uniform und positional.
- **Welche Strategie ist denn nicht positional?**  
Automatenstrategien.
- **Wie kommt man an Automatenstrategien?**  
Per Spielreduktion eines Spiels auf ein Spiel mit positionaler Gewinnstrategie.
- **Was muss dafür bei solchen Spielreduktionen gelten?**  
Hm, man muss für eine Spielreduktion bereits den Speicher und die Transitionsfunktion des Automaten angeben. . .  
**Was muss der Speicher sein, damit man eine Automatenstrategie erhält?**  
Endlich.
- **Gibt es noch andere Arten von Strategien?**  
Man kann noch andere Strategien definieren, aber ich kenne kein Spiel, für die man sie braucht.
- **Dann definieren Sie mal eine.**  
Eine Funktion  $f$  bekommt den gesamten bisherigen Lauf und bildet ihn auf den nächsten Knoten ab.
- **Was gibt es noch zwischen solchen allgemeinen Strategien und Automatenstrategien? Oder meinen Sie die allgemeinen Strategien?**  
Die meinte ich.  
**Dazwischen gibt es noch die berechenbaren Funktionen dieser Art, auch eine interessante Klasse.**
- **Nennen Sie mal ein Beispiel für eine Spielreduktion.**  
Die Reduktion von verallgemeinerten Erreichbarkeitsspielen.
- **Hm, was ist das für ein Spiel?**  
Das wurde in der Vorlesung als Beispiel für Spielreduktionen verwendet. Dabei gibt es mehrere Endzustandsmengen  $F_1, \dots, F_k$ , die alle erreicht werden müssen.
- **Aha. Was für Speicher braucht man da für eine Gewinnstrategie?**

Alle Teilmengen von  $\{1, \dots, k\}$ , um zu speichern welche Mengen bereits erreicht wurden, und wenn eine neue Menge erreicht wird, wird der Speicher entsprechend aktualisiert.

- **Wieviel Speicher braucht man für Büchi-Spiele?** (*Huch, Fangfrage? Nein, Herr Thomas nahm an, ich hätte verallgemeinerte Büchi-Spiele gemeint.*)

Gewinnstrategien für Büchi-Spiele sind positional.

- **Dann reichen doch eigentlich Zahlen von 0 bis  $k$ , so dass man nur hochzählen muss wenn die nächste Menge erreicht wird?**

Nein, bei diesem Spiel muss jede Menge  $F_i$  nur einmal erreicht werden.

**Ach so. Ja, dann reichen Teilmengen der Indexmenge  $\{1, \dots, k\}$ , nicht Teilmengen von Zuständen.** (*Wieder ein Missverständnis ausgeräumt.*)

- **Wie wendet man die Spielreduktion bei anderen Spielen an? Zum Beispiel bei Muller-Spielen.**

Die kann man auf Paritätsspiele reduzieren, mit der LAR-Konstruktion.

- **Wie groß wird dann der neue Spielgraph?**

$n!$

Ja, bzw.  $O(n!)$ .

### 3 Infinite State System Verification

- **Nun zur Verifikation. Da verwendet man ja gerne Logiken. Was für eine Logik kommt Ihnen denn da in den Sinn?**

MSO.

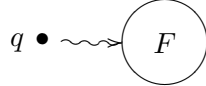
- **Hm, ja, aber damit tut man sich etwas schwer, viele verwenden lieber eine ganz coole Logik. . .**

LTL.

- **Ein typisches Problem ist ja die Erreichbarkeit. Nennen Sie mal ein unendliches Transitionssystem, bei dem die Erreichbarkeit entscheidbar ist.**

Pushdown Graphs. Dort kann man  $\text{pre}^*$  und  $\text{post}^*$  einer Zustandsmenge berechnen.

- **Angenommen, wir haben diese Situation gegeben:**



Ich möchte wissen, ob  $F$  von  $q$  aus erreichbar ist. Wie geht das?

Man berechnet  $\text{pre}^*(F)$  und schaut, ob  $q$  darin enthalten ist.

- **Geht das auch umgekehrt, mit  $\text{post}^*$ ?**

Ja, man berechnet  $\text{post}^*(q)$  und prüft, ob  $F$  da drin liegt.

**Was prüft man?**

Äh, ob der Schnitt mit  $F$  nichtleer ist.

- **Wenn die Konfigurationsmengen  $F$  unendlich groß sind, wie beschreibt man die?**

Mit regulären Sprachen.

- **Angenommen, ich möchte nicht nur einmal nach  $F$ , sondern immer wieder, wie bei einer Büchi-Bedingung. Kann man das auch entscheiden?**

Ich glaube ja, denn man könnte das vielleicht mit einer MSO-Formel ausdrücken und diese dann überprüfen.

- **Hm, das wäre problematisch, das geht wohl nicht so ohne weiteres, wegen [...] Aber vielleicht kann man das mit MSO-Formeln über einer anderen Struktur machen, fällt Ihnen dazu was ein?**

...

**Sie haben in der Vorlesung ein Verfahren kennengelernt, mit dem man neue Strukturen gewinnen kann...**

Ach, Sie meinen Interpretation und Unfolding?

**Ich meine eines davon.**

Unfolding.

**Was erhält man dann?**

Einen Baum, der die möglichen Läufe in dem Pushdown-Graph darstellt.

**Richtig, und in dem Baum kann man mit MSO-Formeln Aussagen über alle Pfade treffen, und sagen dass in jedem Pfad die Bedingung erfüllt ist.**

- **Zu Interpretation und Unfolding: Da gab es einen Satz.**

Wenn eine Struktur MSO-entscheidbar ist, dann sind auch die Strukturen, die sich daraus durch Interpretation und Unfolding ergeben, MSO-entscheidbar.

**Den Begriff MSO-entscheidbar gibt es so nicht, was meinen Sie damit?**

Alle MSO-Sätze sind entscheidbar.

**Richtig, und wie nennt man das auch?**

Theorie.

**Ja, oder auch Model Checking.**

- **Nennen Sie mal eine Struktur, auf der es unentscheidbare Probleme gibt.**

Das Infinite Grid. Das Halteproblem lässt sich darauf reduzieren.

- **Was für ein Problem genau ist denn dort nicht entscheidbar?**

Das Erreichbarkeitsproblem.

**Oh, das ist sehr einfach, das kann ich sehr schnell entscheiden, da brauche ich nur die Koordinaten vergleichen. (*autsch*) Aber etwas anderes ist nicht entscheidbar.**

*(Ich komme nicht drauf)*

- **MSO-Model-Checking ist da nicht entscheidbar. Man kann dort mit MSO, wie Sie sagten, den Lauf einer Turingmaschine ausdrücken. Wie macht man das denn?**

Man drückt mit einer Formel aus, dass die erste Zeile der Startkonfiguration der TM entspricht, und jede folgende Zeile die Nachfolgerkonfiguration der vorherigen Zeile ist...

- **Ja, und dass irgendwann der Endzustand erreicht wird. Wie sieht die MSO-Formel dafür aus? Womit fängt die an?**

Hm, wie gesagt, in der ersten Zeile...

**Man fängt nicht mit der ersten Zeile an.**

Ach so: Es existiert eine Partition.

- **Genau, und was drückt die aus?**

Welche Symbole und Zustände an den entsprechenden Knoten stehen.



- **Im Infinite Grid ist Erreichbarkeit ja entscheidbar, aber können Sie mir Strukturen nennen, in denen das nicht gilt?**

Der Konfigurationsgraph einer Turingmaschine.

*(lacht)* **Das stimmt, eine beliebige Antwort, aber Turingmaschinen zählen mal nicht, die sind zu einfach.**

Infix rewriting systems.

- **Ja, die kann man nehmen, sind aber sehr nah dran an TM. Noch andere?**

$n$ -Counter-Systems.

- **Ja, stimmt, aber die sind ja auch universell. Noch mehr? So in Richtung Automaten?**

Ich komme nicht drauf.

**Es gibt da noch die CFSMs.<sup>2</sup>**

- **Kommen wir zurück zu  $\text{pre}^*$  und  $\text{post}^*$ . Welcher Satz war das, den man dafür benutzt?**

Der Satz von Büchi.

- **Was sagt der aus?**

Wenn eine Konfigurationsmenge  $R$  regulär ist, sind auch  $\text{pre}^*(R)$  und  $\text{post}^*(R)$  regulär.

- **Können Sie noch kurz etwas dazu sagen, wie man diese Mengen berechnet?**

Man verwendet einen  $P$ -Automaten zur Beschreibung der Konfigurationsmenge und wendet den Saturierungsalgorithmus an.

## 4 Rekursionstheorie

- **Wir haben die Begriffe rekursiv aufzählbar und rekursiv behandelt. Bei den rekursiven Funktionen gibt es noch die primitiv rekursiven Funktionen. Das Kleenesche Normalform-Theorem stellt da einen Zusammenhang her, können Sie den angeben?**

$f$  rekursiv  $\iff f(\bar{x}) = U(\mu y T e \bar{x} y)$ . Jede rekursive Funktion ist so darstellbar, wobei  $T$  primitiv rekursiv ist, so dass man nur einen  $\mu$ -Operator braucht.

---

<sup>2</sup>Aber die sind eigentlich auch universell...

- **Wie kann man rekursiv aufzählbare Mengen charakterisieren? Am besten so dass es so ähnlich aussieht wie das da.**

$W_x \dots$

$x$  ist nicht so geeignet.

Ja,  $e$  ist besser:  $x \in W_e \iff \exists y T e x y$

- **Da gab es auch die Begriffe Reduktion und Vollständigkeit. Was für Reduktionen haben wir betrachtet?**

$m$ -Reduktion, 1-Reduktion,  $tt$ -Reduktion und Turing-Reduktion.

- **Was bedeutet es, wenn eine Menge 1-vollständig ist? Wenn wir mal von rekursiv aufzählbaren Mengen ausgehen.**

Die Menge ist rekursiv aufzählbar, und jede rekursiv aufzählbare Menge ist auf sie 1-reduzierbar.

- **Gibt es überhaupt 1-vollständige Mengen?**

Ja,  $K$  ist 1-vollständig.

- *(schreibt)*  $A \leq_T K \iff A$  rek. aufz. Ist das so richtig?

Nein.

- **Welche Richtung ist falsch?**

Links nach rechts.

- **Ändern Sie das mal, dass es richtig wird.**

$A \leq_T K \iff A \in \Delta_2^0$

- **Schreiben Sie mal dazu was  $\Delta_2^0$  ist. Mit anderen griechischen Buchstaben.**

$\Delta_2^0 = \Sigma_2^0 \cap \Pi_2^0$

- **Können Sie eine Menge nennen, die in  $\Pi_2^0$  liegt, und auch für  $\Pi_2^0$  vollständig ist?**

$D_2$ .<sup>3</sup>

Ach ja, sowas haben wir auch behandelt. Ich meinte eher was mit Quantoren und so. Sagen wir,  $A$  liegt in  $\Pi_2^0$ .

Ergänzen Sie mal:  $x \in A \iff$

$x \in A \iff \forall y \exists z R x y z$ , wobei  $R$  rekursiv ist.

---

<sup>3</sup>Hinterher ist mir aufgefallen, dass das falsch ist,  $D_2 \in \Sigma_2^0 \setminus \Pi_2^0$ . Richtig wäre  $\overline{D_2}$ .

- **Nennen Sie mal eine konkrete Menge von der Form. Ich denke da so an Aussagen über Definitionsbereiche...**  
Die Menge der totalen rekursiven Funktionen.  
**Hmm, für alle Argumente gibt es eine Schrittzahl so dass die Funktion hält, ja genau.**
- **Nochmal zu  $A \leq_T K \iff A \in \Delta_2^0$ . Wie heißt der Satz, nach dem das gilt?**  
Kleene-Post.

## 5 Compilerbau

- **Mal ein paar Fragen zur Syntaxanalyse. Wir, das heißt, Sie haben ja die Bottom-Up-Analyse betrachtet. Welche Grammatiken verwendet man dafür?**  
LR( $k$ ).
- **Welche haben Sie kennengelernt?**  
LR(0) und LR(1).
- **Welche ist Ihnen am liebsten?**  
LR(0).
- **(lacht) Wie arbeitet denn der BU-Analyseautomat bei LR(1)?**  
Der Automat entscheidet anhand des Lookahead-Symbols und der Kellerspitze, ob ein Shift- oder Reduce-Schritt durchgeführt werden muss.
- **Bei LR(0)-Grammatiken verwendet man LR(0)-Mengen, was können Sie dazu sagen?**  
(Definition von LR(0)-Auskünften hingeschrieben und erklärt)
- **Ja, aber welche Rolle spielen sie beim Analyseautomat? Bei der Eingabe, der Ausgabe, als Kellularphabet...?**  
Die LR(0)-Mengen werden als Kellularphabet verwendet.
- **So ein Analyseautomat für LR(0) muss ja nach einem deterministischen Verfahren arbeiten...**  
Dafür gibt es die goto-Funktion, die abhängig von der LR(0)-Menge auf der Kellerspitze und der Eingabe eine neue Menge auf den Keller legt, und die action-Funktion, die die Shift- bzw. Reduce-Entscheidung liefert.

- **Kann man für eine gegebene Grammatik entscheiden ob sie in LR(0) ist?**  
Ja... *\*überleg\** ach ja, die action-Funktion muss eindeutig sein, das heißt, es darf keine Konflikte geben.
- **Kommen wir zu LL(1). Wie entscheidet man, ob  $G$  in LL(1) ist?**  
Man berechnet die Lookahead-Mengen, und prüft, ob sie für Regelpaare  $A \rightarrow \beta|\gamma$  disjunkt sind.
- **Sind alle kontextfreien Grammatiken in LL(1)?**  
Nein, zum Beispiel sind linksrekursive Grammatiken nicht in LL(1).
- **Was kann man da tun?**  
Man kann sie in die Greibach-Normalform bringen, bzw. in manchen Fällen die Linksrekursion direkt auflösen.
- **Nun kann man das ja mit allen kontextfreien Grammatiken machen, deswegen heißt es ja Normalform. Sind dann alle so umgewandelten Grammatiken in LL(1)?**  
Nein.
- **Gut, noch kurz zu Attributgrammatiken und Zirkularitäten. Was ist eine Zirkularität?**  
Zirkularität bedeutet, dass ein Attribut in einem Ableitungsbaum von sich selbst abhängt.
- **Kann das auftreten, wenn es nur synthetische Attribute gibt?**  
Nein...  
**... oder nur inherite Attribute?**  
Nein, es muss sowohl Abhängigkeiten synthetischer von inheriten Attributen geben, als auch umgekehrt.  
**Gut, dann hören wir mal hier auf, wir haben schon fast 50 Minuten rum.**

## Fazit

Die Note ist 1,7 – dafür, dass ich nur so wenig Zeit zum Lernen hatte, kann sich das sehen lassen. Ich konnte zwar den meisten Stoff, er war aber noch nicht sehr gefestigt. Das hat sich dann so geäußert, dass ich an vielen Stellen gezögert habe und überlegen musste. Ich kann bestätigen, dass die Prüfungsatmosphäre sehr angenehm war, und die Hilfestellungen waren in

der Tat hilfreich. Zwar hat Herr Thomas mir für den kurzen Compilerbau-Teil eine 1,0 gegeben, aber auch in der Gesamtnote berücksichtigt, dass er nicht so viel Stoff geschafft hat wie er vorhatte. Ansonsten negativ in die Note eingegangen sind die falsche Büchi-Sprache, die Erreichbarkeit im Infinite Grid und die CFSMs, die mir nicht eingefallen sind.

Falls mich der Ehrgeiz packt und ich eine Verbesserungsprüfung mache, findet ihr hier in spätestens vier Monaten eine erweiterte Version. :-)

Viel Erfolg bei euren Prüfungen!