

Gedächtnisprotokoll - Vordiplomsprüfung Mathe für Informatiker 1 und Grundlagen der theoretischen Informatik

Christian Stade-Schuldt (stade@inf.fu-berlin.de)

26. Mai 2004

Prüfer: Dr. Klaus Kriegel

Beisitzer: ?

Prüfungsdatum: 25.05.2004

Protokoll erstellt am: 26.05.2004

Note: 2,7

1 Vorbereitung

Die Vorbereitung hat knapp länger als eine Woche gedauert. Ich habe die Definitionen auswendig gelernt und die Beweise versucht nachzuvollziehen. Als Lektüre habe ich zur Vorbereitung verwendet:

- Meine Aufzeichnungen aus den Vorlesungen zu MafI1 und GTI.
- Theoretische Informatik kurzgefasst von Uwe Schöning

2 Atmosphäre

Herr Kriegel hat es sehr gut geschafft eine angenehme Atmosphäre zu schaffen. Die gesamte Prüfung hatte den Charakter eines Gesprächs über mathematische Sachverhalte. Wenn man mal nicht weiter weiß, gibt er gute Hinweise. Allerdings fragt er auch gerne mal nach, wenn man ein wenig unsicher ist.

3 Ablauf der Prüfung

Was sind die Binomialkoeffizienten und was gibt er an?

Der Binomialkoeffizient:

$$\binom{n}{k} = \frac{n!}{(n-k)!k!} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Mit ihm berechnet man die Anzahl der k-elementigen Untermengen einer n-elementigen Menge.

Wie kommt man auf die erste Formel?

Ich habe ihm darauf erklärt das sich diese über die k-Permutationen

$$n \cdot (n-1) \cdot \dots \cdot (n-k+1) = n$$

geteilt durch $k!$ herleiten lässt.

Wie kommt man auf die zweite (rekursive) Formel?

Sei M eine n-elementige Menge. Wir betrachten nun ein Element $p \in M$. Die Menge $M \setminus \{p\}$ hat $n-1$ Elemente. Jede k-elementige Untermenge, die p beinhaltet, enthält $k-1$ Elemente aus $M \setminus \{p\}$. Somit gibt es $\binom{n-1}{k-1}$ solcher Teilmengen. Jede k-elementige Untermenge, die p nicht beinhaltet, enthält k Elemente aus $M \setminus \{p\}$. Es gibt also $\binom{n-1}{k}$ solcher Teilmengen. Damit bildet die Summe $\binom{n-1}{k-1} + \binom{n-1}{k}$ die Anzahl k-elementigen Mengen von M .

Was ist der Unterschied zwischen einer booleschen Formel und einer booleschen Funktion?

Boolesche Formeln: Es gibt eine Menge p_1, p_2, \dots, p_n von Variablen, die Primformeln genannt werden. Zusammengesetzte Formeln werden auch durch folgenden induktiven Prozess definiert:

- Für alle Formeln α und β sind auch $\alpha \wedge \beta$ und $\alpha \vee \beta$ Formeln.
- Für jede Formel α ist $\neg\alpha$ eine Formel

Boolesche Funktion: Eine boolesche Funktion ist eine Abbildung $\{0,1\}^n \rightarrow \{0,1\}$ dargestellt durch eine Formel oder Wertetabelle.

Wie kommt man von einer Wertetabelle auf eine einen zugehörigen Term?

DNF oder KNF bilden. Ich habe dann noch erklärt was eine KNF bzw. DNF ist und wie man es an einem konkreten Beispiel macht.

Wie kann man feststellen, ob eine Funktion immer wahr ist?

Testen aller Belegungen (bei n Variablen 2^n Zeilen) oder mittels Resolutionskalkül prüfen, ob die Negierung unerfüllbar, also eine Kontradiktion ist.

Wie funktioniert das Resolutionskalkül?

Formel ist in KNF gegeben. Man bildet Resolventen nach dem Resolutionslemmas und wenn die leere Klausel ableitbar ist, dann ist die Formel nicht erfüllbar.

Können Sie das Resolutionslemma beweisen?

Es existieren zwei Klauseln K_1 und K_2 der Klauselmenge K_A und ein Literal $l \in K_1$ und $\bar{l} \in K_2$. Weiterhin sei K_A erfüllt durch die Belegung ω .

Wenn nun $\omega(l) = 0$, dann muss eine Literal $l_2 \in K_1$ existieren, so dass $\omega(l_2) = 1$, da in einer KNF jede Klausel erfüllbar sein muss.

Dann wenden wir uns mal dem zweiten großen Thema zu. In TI hatten wir ja auch mit dem Erfüllbarkeitsproblem zu tun...

Ich habe ihm die Definition des SAT-Problems gesagt und das es NP-vollständig ist.

Wann ist denn eine Sprache NP-vollständig?

Eine Sprach liegt in NP, wenn sie in Polynomialzeit verifizierbar ist. Eine Sprache ist NP-hart, wenn für alle Sprachen $L \in NP$ gilt $L \leq_p A$. Eine Sprache ist NP-vollständig, wenn sie in NP liegt und NP-hart ist.

Was bedeutet polynomialzeit-reduzierbar?

L_1 ist polynomialzeit-reduzierbar auf L_2 falls eine Funktion f existiert, so dass gilt: $w \in L_1 \Leftrightarrow f(w) \in L_2$.

Wenn es eine Möglichkeit gibt, eine NP Problem in polynomieller Zeit zu lösen. Was hätte dies zur Folge.

Dann wäre auch das SAT Problem in polynomieller Zeit lösbar. Man überführt dazu einfach das SAT-Problem in polynomieller Zeit in das gelöste und löst dieses dann. Die Komposition ist immer noch in polynomieller Zeit.

Wie sind die Regulären Sprachen definiert?

Reguläre Sprachen sind durch eine reguläre Grammatik beschreibbar.

Welche Äquivalenzen gibt es dazu?

Folgende Aussagen sind äquivalent:

- L ist durch einen DFA beschreibbar
- L ist durch einen NFA beschreibbar
- L ist durch reguläre Ausdrücke beschreibbar
- Der Index der Nerode-Relation ist endlich

Beschreiben Sie das Pumping-Lemma...

Ist L eine reguläre Sprache, dann gibt es ein $n \in \mathbb{N}$, so dass für jedes $z \in L$ mit $|z| \geq n$ eine Zerlegung $z = uvw$ mit

1. $|uv| \leq n$
2. $|v| \geq 1$
3. $uv^i w$ für alle $i \in \mathbb{N}$

Beweisen Sie das Pumping-Lemma...

Da L regulär ist, gibt es einen DFA M, der L akzeptiert. Wir wählen für n die Zahl der Zustände von M : $n = |Z|$. Sei nun x ein beliebiges Wort der Länge $\geq n$, das der Automat akzeptiert. Beim Abarbeiten von x durchläuft der Automat $|x| + 1$ Zustände (den Startzustand mitgezählt). Da $|x| \geq n$ können diese Zustände nicht alle verschieden sein (Schubfachsatz). Also muss der DFA eine Schleife durchlaufen.

Hier hat Herr Kriegel dann abgebrochen und meinte das ja auch die Zeit schon längst rum sei.

4 Fazit

Ich musste kurz raus und danach wurde mir die Note mitgeteilt. Man war der Meinung, dass ich mich gut vorbereitet hätte, jedoch durch meine Aufregung oft zu knapp geantwortet hatte und auch manchmal den einen oder anderen Fakt durcheinander gewürfelt hätte. Ich bin jedenfalls mit der Note zufrieden.

Ich wünsche allen den die Prüfung noch bevorsteht viel Erfolg Bei Rückfragen und/oder Feedbacks schreibt einfach eine E-Mail an stade@inf.fu-berlin.de oder besucht mich unter <http://www.tafkas.org>