1

28.06.2018

Wahrscheinlichkeitsrechnung und Algorithmik

Sommersemester 2018

6. Übung

Zum Local Lemma (Algorithmus f. SAT)

d = # Nachbarn einer Klausel ~~ohne~~ inklusive selbst

$$d \lneqq \frac{2^k}{e}$$

Algorithmus: Setze bei irgendeiner falschen Klausel
<u>alle</u> Variablen zufällig neu!

Lauf d. Algorithmus: $C_1, C_2, \ldots, C_k, C_{k+1}, \ldots$

(Schluß, wenn alle wahr sind)

$\underline{\underline{C_1}}, C_2, \ldots$

C_j

- Belegung ist zufällig, eine von 2^n Bel.

α_1 = Belegung nach 1. Schritt; bleibt zufällig, da alle einer Klausel zufällig neu.

Satz: Bei gegebenem F mit $d \lneq \frac{2^k}{e}$, $d \leq \left\lfloor \frac{2^k}{e} \right\rfloor$

ist Prob[Algo rechnet mindestens $Q \cdot M$ Schritte] $\to 0$

für $Q > C' \cdot \log M$

($M = \#$ Klauseln von F)

Beweis: Rechnung $C_1, C_2, \ldots, C_k, C_{k+1}, \ldots$ mind. $Q \cdot M$ Schritte lang.

$\alpha_0 \quad \alpha_1 \quad \alpha_2 \ldots$

Anfangs-belegung

Bsp.: $\underbrace{(x_1 \vee x_2 \vee x_3)}_{C_1} \wedge \underbrace{(\neg x_3 \vee x_4 \vee x_5)}_{C_2} \wedge \underbrace{(x_5 \vee x_6 \vee x_7)}_{C_3}$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
α_0	0	0	0	0	0	0	0
α_1	0	0	1				
α_2			0	0	0		
α_3	0	0	1				
$\vdots$							

Beispielrechnung: $C_1, C_2, C_1, C_2, \ldots C_3$

~~hängt~~ hängt direkt und indirekt von vorherigen Ziehungen ab.

Haben wir Rechnung der Länge $\geq Q \cdot M$, dann Rechnung der Art

$A|||.|, C', ||.||.|, C', ||||.|.|, C', ||.|| |, C', ||.|.|, C'$

disjunkt von der rest. Rechnung

$\geq Q$ mal die Klausel C'

Haben Folge von $q \geq Q$ ~~Klau~~ Stück, deren Züge aufeinander aufbauen!

⇝ für jede Klausel in der Folge sind k Bits gesetzt.

⇝ genau $q \cdot k$ Bits sind fest gesetzt (damit C' q-mal vorkommt)

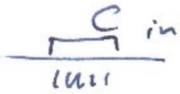
(Wo sind die Bits in der Tabelle?)

Für so ein Stück der Rechnung, das alle Klauseln beinhaltet, die mittelbar und ~~unter~~ unmittelbar sich auswirken, gilt:

Es lassen sich genau die Bits aus der Tabelle angeben, die die Rechnung <u>ermöglichen</u>.

Haben wir so eine Rechnung (dieses Stück) der Länge $= q$ dann sind $k \cdot q$ Bits der Tabelle bestimmt.

(d.h. eine Möglichkeit von $2^{k \cdot q}$ muß auftreten.)

$q = 1$ C  in α_0 ist falsch gesetzt.

Rechnung ist $\geq Q \cdot M$, dann haben wir ein Stück (im obigen Sinne) das aus $q \geq Q$ Klauseln besteht.

- W-Keit eines festen Stückes $= \frac{1}{2^{kq}}$
- Prob[Rechnung der Größe $\geq Q \cdot M$]

$\leq$ Prob[Es gibt Stück der Länge q für ein $q \geq Q$]

$$\leq \sum_{q \geq Q} \text{Prob[Es gibt ein Stück der Länge } q \text{]}$$

Prob[Es gibt Stück der Länge q]

$$\leq \sum_{\substack{\text{Mögl.} \\ \text{Stücke } S \\ \text{Länge } q}} \text{Prob[} S \text{ tritt auf]}$$

$$= \underbrace{(\#\text{Stücke d. Länge } q)}_{} \cdot \frac{1}{2^{k \cdot q}}$$

Suchen Abschätzung dafür:

1. Versuch: M^q ist sicherlich Abschätzung

$M^q \cdot \frac{1}{2^{k \cdot q}} \to \infty$ bei M groß

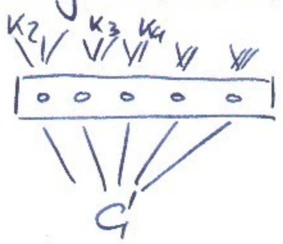
Tatsächlich gilt:

#Stücke der Länge q ist

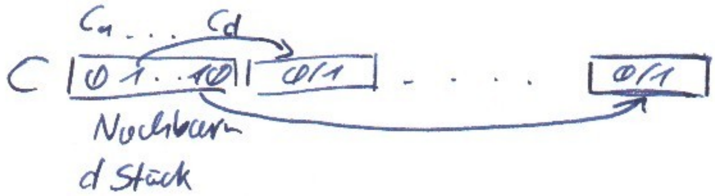
$$\leq (\underbrace{d\cdot e}_{\text{Konst.}})^q$$

Wie entsteht ein Stück?

1. Nehmen die letzte Klausel (= Wurzel)
(M Möglichkeiten)



K_1 aus d
Stück



Stück $\mapsto$ 0-1-Folge mit $q-1$ Einsen und
$q\cdot d$ Positionen insgesamt.

$\binom{q\cdot d}{q-1}$ Bitfolgen

$$\# \text{Stücke d. L. } q. \leq M\cdot\binom{q\cdot d}{q-1} \leq M\cdot\binom{q\cdot d}{q} \leq M\cdot\left(\frac{q\cdot d\cdot e}{q}\right)^q = M\cdot(d\cdot e)^q$$

Also:

$$\sum_{q \geq Q} \frac{1}{2^{kq}} \cdot M \cdot (d \cdot e)^q$$

$$= \sum_{q \geq Q} \underbrace{\left(\frac{d \cdot e}{2^k}\right)}_{\lneq 1}^{q} \cdot M$$

$$= \left(\frac{d \cdot e}{2^k}\right)^Q \cdot \left(\frac{1}{1 - \frac{d \cdot e}{2^k}}\right) \cdot M$$