Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – Hausaufgabenblatt 12

- Die Aufgaben werden in folgender Reihenfolge korrigiert: H12.1, H12.2
- Die Knobelaufgabe bitte separat auf Moodle abgeben. Sie wird korrigiert.
- Die genauen Details, wie wir die Eingabe einer Sprache kodieren, sind für uns nicht relevant. Es ist z.B. erlaubt, $\overline{\mathsf{SAT}} = \{\text{Formel } F : F \text{ ist nicht erfüllbar}\}$ zu schreiben, obwohl eigentlich auch Wörter, die überhaupt keine Formel kodieren, in $\overline{\mathsf{SAT}}$ enthalten wären. Eine Ausnahme bildet nur, wenn die Wahl der Kodierung die Länge der Eingabe signifikant beeinflussen würde (z.B. unäre/binäre Darstellung von Zahlen).

Aufgabe H12.1. (*Satisfied?*) 4+4+4+4 Punkte

Zwei aussagenlogische Formeln über denselben Variablen sind *äquivalent*, wenn sie dieselben erfüllenden Belegungen haben, d.h. jede Belegung erfüllt beide Formeln oder keine.

Nehmen Sie $\mathsf{P} \neq \mathsf{NP}$ an. Entscheiden Sie unter dieser Annahme, welche der folgenden Probleme in P und welche NP-vollständig sind.

Wenn das Problem in P liegt, beschreiben Sie einen polynomiell zeitbeschränkten Algorithmus und begründen Sie die Korrektheit und polynomielle Beschränktheit des Algorithmus kurz.

Wenn das Problem NP-vollständig ist, geben Sie eine Reduktion von $\mathsf{3KNF\text{-}SAT}$ an. Die Reduktionsfunktion muss präzise definiert sein. Die Korrektheit der Reduktion müssen Sie aber nur skizzieren und nicht formal beweisen.

(a) **Gegeben:** Eine KNF-Formel F.

Problem: Ist $\neg F$ erfüllbar?

(b) **Gegeben:** Zwei KNF-Formeln F und G über denselben Variablen.

Problem: Sind F und G nicht äquivalent?

(c) **Gegeben:** Eine KNF-Formel F.

Problem: Gibt es eine erfüllbare Belegung, die höchstens 3 oder mindestens $v-3$ Variablen wahr macht, wobei v die Anzahl der Variablen von F ist?

(d) **Gegeben:** Eine KNF-Formel F mit einer geraden Anzahl von Variablen.

Problem: Gibt es eine erfüllbare Belegung, die genau die Hälfte der Variablen von F wahr macht?

Lösungsskizze.

(a) Wenn $F = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n_i} L_{i,j}$, dann gilt $\neg F \equiv \bigvee_{i=1}^{n} \bigwedge_{j=1}^{n_i} \overline{L_{i,j}}$ mit $\overline{L_{i,j}} := \neg x$ falls $L_{i,j} = x$ und $\overline{L_{i,j}} := x$ falls $L_{i,j} = \neg x$.

Wir definieren $F_i := \bigwedge_{j=1}^{n_i} \overline{L_{i,j}}$. Die Formel $\neg F \equiv \bigvee_{i=1}^{n} F_i$ ist erfüllbar genau dann wenn mindestens ein F_i erfüllbar ist.

Wir gehen alle F_i hintereinander durch. Für jedes F_i prüfen wir, ob es erfüllbar ist. Das ist der Fall gdw. es keine Variable x gibt, sodass sowohl x als auch $\neg x$ in F_i vorkommen. Dies kann für jede Variable in linearer Zeit geprüft werden.

(b) Sei φ eine 3KNF-Formel, sei x eine Variable. Wir konstruieren die Formeln $F \coloneqq \varphi$ und $G \coloneqq x \wedge \neg x$.

G ist unerfüllbar. Somit gilt: F ist erfüllbar gdw. $F \not\equiv G$.

(c) Es gibt $2(1 + v + \frac{v(v-1)}{2} + \frac{v(v-1)(v-2)}{6})$ Belegungen (polynomiell viele), die die Bedingung erfüllen.

Für jede solche Belegung können wir in linearer Zeit überprüfen, ob sie die Formel erfüllt.

(d) Gegeben eine 3KNF-Formel φ mit n Variablen, konstruiere die Formel $F \coloneqq \varphi \wedge (y_1 \vee \cdots \vee y_n \vee \neg y_1)$, wobei $y_1, \dots, y_n$ neue Variablen sind.

Wenn φ erfüllbar ist mit einer Belegung mit k wahren Variablen, dann erweitere die Belegung mit $y_1 \mapsto 1, \dots, y_{n-k} \mapsto 1, y_{n-k+1} \mapsto 0, \dots, y_n \mapsto 0$.

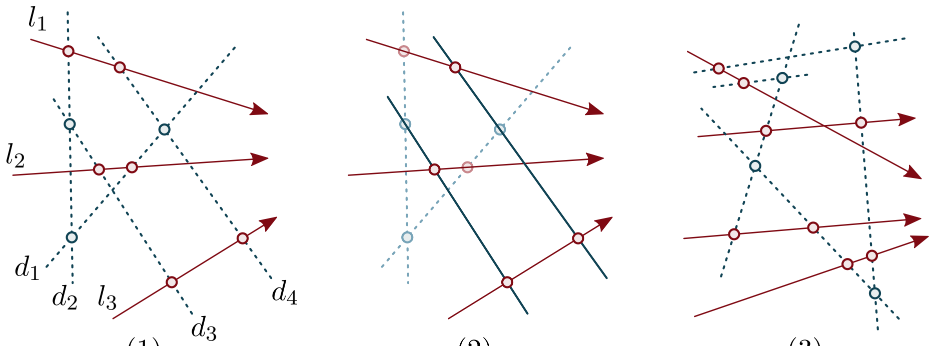
Wenn φ unerfüllbar ist, dann ist auch F unerfüllbar.

Aufgabe H12.2. (*Laser-focused*) 4+3+4+6 Punkte

Nachdem er sich einige Wochen lange verdeckt gehalten hatte, hatten die meisten den berüchtigten Superschurken Dr. Evilsparza schon vergessen. Nun ist er aber wieder aufgetaucht, mit einem diabolischen Plan, die Weltherrschaft an sich zu reißen! Er hat eine Reihe von Superlasern aufgestellt, mit denen er direkt auf den THEO-Hausaufgabenserver zielt. Ohne Musterlösungen werden sicher alle Teilnehmer durchfallen, sodass es keine neuen Informatikabsolventen gibt, und alle auf das IT-Consultingunternehmen „Evillest Evilness Software *Inc*“ angewiesen sind. Zum Glück hat der Lehrstuhl für KI (Klausurintegrität) einen Plan.

Spezielle, mit Spiegeln ausgestattete Drohnen können losgeschickt werden, um die Superlaser zu ihrer Quelle zurückzuschicken und so zu zerstören. Auf einer Karte sind bereits die Schusslinien der Laser eingezeichnet, die möglichen Flugbahnen der Drohnen, die Punkte, an denen eine Drohne den Laserstrahl kreuzen würde, und die Punkte, an denen sich die Flugbahn zweier Drohnen kreuzen würde. Letzteres würde zum Absturz beider Drohnen führen und muss unbedingt vermieden werden.

Ihre Aufgabe ist es nun, herauszufinden welche Drohnen gestartet werden müssen, sodass sich keine zwei Drohnen kreuzen und alle Laser zerstört werden.



(1) (2) (3)

Formal definieren wir das Klausurrettungsproblem (KRP) wie folgt:

Eingabe: Mengen an Flugbahnen L, D und Schnittpunkte $S \subseteq L \times D \cup D \times D$

Ausgabe: Existiert $M \subseteq D$ mit $M \times M \cap S = \emptyset$ und $\{l\} \times M \cap S \neq \emptyset$ für alle $l \in L$?

In der Zeichnung finden Sie eine Instanz des KRP (1), mit

$$L = \{l_1, l_2, l_3\}$$
$$D = \{d_1, d_2, d_3, d_4\}$$
$$S = \{(l_1, d_2), (l_1, d_4), (l_2, d_3), (l_2, d_1), (l_3, d_3), (l_3, d_4), (d_1, d_2), (d_2, d_3), (d_1, d_4)\}$$

Außerdem abgebildet sind eine Lösung dieser Instanz (2), mit $M = \{d_3, d_4\}$, und eine weitere, unlösbare Instanz (3). Die Laser sind jeweils als Pfeile gezeichnet und die möglichen Flugbahnen der Drohnen als gestrichelte Linien.

Achtung: Die graphische Darstellung dient nur der Übersichtlichkeit. Die Eingabe der Instanz umfasst nur die Mengen, wie oben. Beachten Sie insbesondere, dass nicht jeder Schnittpunkt von zwei Graden bedeutet, dass sich die Drohnen bzw. Laser da kreuzen würden.

(a) Konstruieren Sie einen polynomiell beschränkten Verifikator für KRP (s. Def. 6.7). Beschreiben Sie insbesondere, wie ein Zertifikat für eine Instanz $w \in$ KRP aussieht, und wie dieses Zertifikat in polynomieller Zeit überprüft werden kann.

In der Vorlesung haben wir das NP-vollständige Problem 3COL (3-Färbbarkeit) kennen gelernt. (Anm.: Wir zeigen dort nicht, dass es NP-vollständig ist.) Unser Ziel ist jetzt, 3COL auf KRP zu reduzieren.

(b) Zunächst wollen wir eine Instanz konstruieren, mit der wir die Farbe eines einzelnen Knotens v modellieren können. Geben Sie eine Instanz vom KRP an, die Drohnen v_1, v_2, v_3 enthält und deren einzige Lösungen $M = \{v_1\}$, $M = \{v_2\}$ und $M = \{v_3\}$ sind. (Sie müssen dazu weitere Drohnen und/oder Laser verwenden.) Begründen Sie kurz, wieso Ihre Instanz korrekt ist.

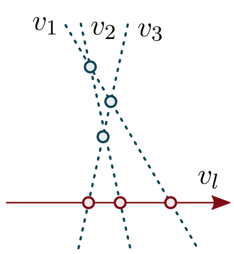
Anmerkung: So eine Konstruktion wird allgemein auch als Gadget bezeichnet.

(c) Erweitern Sie nun Ihre Konstruktion aus (b), um den Graphen mit Knoten $\{u, v\}$ und der Kante $\{u, v\}$ zu modellieren. Verwenden Sie dazu zwei Kopien ihrer Lösung aus (b), und erweitern die Instanz geeignet, sodass es 6 Lösungen gibt, und keine davon sowohl u_i als auch v_i beinhaltet, für $i \in \{1, 2, 3\}$.

(d) Zeigen Sie 3COL $\leq_p$ KRP.

Lösungsskizze. **(a)** Sei (L, D, S) eine Instanz vom KRP. Als Zertifikat verwenden wir M. Da $|M| \leq |D|$, ist die Länge von M polynomiell in der Länge der Instanz. Wir können überprüfen, dass M ein gültiges Zertifikat ist, indem wir feststellen, ob $M \times M \cap S = \emptyset$ und $\{l\} \times M \cap S \neq \emptyset$ für alle $l \in L$ gilt. Die Mengen $M, M \times M, S$ und L sind nur polynomiell groß, also lässt sich dies in polynomieller Zeit auf die offensichtliche Art und Weise lösen.

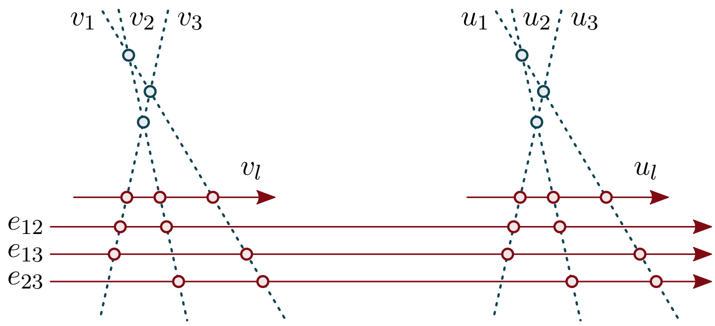
(b) Unsere Konstruktion sieht folgendermaßen aus:



Formal verwenden wir also drei Drohnen $D := \{v_1, v_2, v_3\}$ und einen Laser $L := \{v_l\}$. Die Schnittpunkte sind $S := \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_1, v_l), (v_2, v_l), (v_3, v_l)\}$.

Die Farben werden über die Drohnen modelliert, für jede Farbe gibt es eine Drohne. Die Drohnen kollidieren paarweise, sodass keine zwei Farben gleichzeitig gewählt werden können, und sie kollidieren jeweils mit dem Laser, sodass mindestens eine Farbe gewählt werden muss.

(c)



(d) Wir erstellen eine Kopie des Gadgets aus (b) für jeden Knoten v, wobei wir die Drohnen v_1, v_2, v_3 nennen, und den Laser v_l. Für jede Kante $e = (u, v)$ erstellen wir nun drei Laser e_{12}, e_{13}, e_{23}, mit Schnittpunkten (v_i, e_{ij}) und (v_j, e_{ij}) für alle e_{ij}. Dies lässt sich offensichtlich in polynomieller Zeit machen. Wir zeigen nun „KRP lösbar $\Leftrightarrow$ 3COL lösbar“:

„$\Rightarrow$“ Sei M die Lösung des KRP. Für jeden Knoten v ist nach Konstruktion unseres Gadgets nun genau eine der Drohnen v_1, v_2, v_3 in M. Wir wählen nun eine entsprechende Färbung der 3COL Instanz. Angenommen die Färbung wäre nicht korrekt, es gibt also eine Kante $e = (u, v)$ sodass sowohl u als auch v die gleiche Farbe haben. Nach Wahl der Färbung gibt es dann $i, j \in \{1, 2, 3\}$ mit $i \neq j$ (also die beiden anderen Farben), sodass $v_i, v_j, u_i, u_j \notin M$ gelten. Dann wird der Laser e_{ij} nicht aufgehalten, da dies die einzigen Drohnen sind, die ihn kreuzen. Also muss die Färbung korrekt sein.

„$\Leftarrow$“ Sei $f : V \to \{1, 2, 3\}$ eine Färbung. Wir wählen nun $M := \{v_{f(v)} : v \in V\}$. Jeder Laser v_l wird also aufgehalten. Für die Laser e_{ij}, wobei $e = (u, v)$ eine Kante ist, gilt $i \neq j$ und $f(u) \neq f(v)$. (Letzteres, da f eine Färbung ist.) Somit folgt $\{i, j\} \cap \{f(u), f(v)\} \neq \emptyset$ nach Schubfachprinzip, und $u_{f(u)}$ oder $v_{f(v)}$ decken diesen Laser ab.

Quizaufgabe H12.3. (*Eine Tautologie ist eine Tautologie*) unkorrigiert (8 Punkte)

Doras Freundin Bijanka (eine Kaffeemaschine) geht es gerade nicht so gut. Sie sucht schon seit Stunden nach einer Lösung für eine Formel, die Theo ihr gesagt hat, kann trotz ihres hochmodernen Prozessors aber keine finden. (Dora vermutet, dass Theo mal wieder versucht, seine Hausaufgaben nicht selbst machen zu müssen.) Sie möchte Bijanka trösten, indem sie sich eine Formel ausdenkt, für die es ganz leicht ist, eine Lösung zu finden! Die NP-Probleme sind dafür schlecht, da es für sie schwierig ist, Lösungen zu finden – also denkt sich Dora eine neue Klasse aus, die sie coffee-NP (kurz coNP) nennt.

Wir definieren coNP $:= \{\overline{L} : L \in \text{NP}\}$. Wir nennen eine aussagenlogische Formel F *Tautologie*, wenn F für jede Belegung erfüllt ist. Das Problem TAUTOLOGY ist nun

Eingabe: Aussagenlogische Formel F
Ausgabe: Ist F eine Tautologie?

(a) Beweisen Sie $\overline{\mathsf{SAT}} \leq_p \mathsf{TAUTOLOGY}$.

(b) Seien L_1, L_2 Sprachen. Zeigen Sie $L_1 \leq_p L_2 \Leftrightarrow \overline{L_1} \leq_p \overline{L_2}$.

(c) Seien L_1, L_2, L_3 Sprachen. Zeigen Sie $L_1 \leq_p L_2 \wedge L_2 \leq_p L_3 \Rightarrow L_1 \leq_p L_3$.

(d) Zeigen Sie, dass $\mathsf{TAUTOLOGY}$ coNP-hart ist, also $L \leq_p \mathsf{TAUTOLOGY}$ für jedes $L \in \mathsf{coNP}$.

Lösungsskizze. **(a)** Eine Formel F ist (nach Definition) erfüllbar, gdw. es eine Belegung σ gibt, sodass $\sigma(F) = 1$. Für $F \notin \mathsf{SAT}$ gilt also $\sigma(F) = 0$ für *alle* Belegungen σ. Die Formel $\neg F$ ist dann eine Tautologie. Insgesamt erhalten wir

$$F \in \overline{\mathsf{SAT}} \Leftrightarrow (\forall\sigma\colon \sigma(F) = 0) \Leftrightarrow (\forall\sigma\colon \sigma(\neg F) = 1) \Leftrightarrow \neg F \in \mathsf{TAUTOLOGY}$$

Unsere Reduktion negiert also einfach die Formel, was offensichtlich in polynomieller Zeit geht.
(b) Wenn $L_1 \leq_p L_2$, dann gibt es eine polynomiell berechenbare Funktion f, die L_1 auf L_2 reduziert. Wir können die gleiche Funktion verwenden, um $\overline{L_1}$ auf $\overline{L_2}$ zu reduzieren:

$$w \in \overline{L_1} \;\Leftrightarrow\; w \notin L_1 \;\Leftrightarrow\; f(w) \notin L_2 \;\Leftrightarrow\; f(w) \in \overline{L_2}$$

Damit haben wir $L_1 \leq_p L_2 \Rightarrow \overline{L_1} \leq_p \overline{L_2}$ gezeigt. Indem wir diese Aussage auf die rechte Seite anwenden, erhalten wir

$$\overline{L_1} \leq_p \overline{L_2} \;\Rightarrow\; \overline{\overline{L_1}} \leq_p \overline{\overline{L_2}} \;\Leftrightarrow\; L_1 \leq_p L_2$$

(c) Sei f die Reduktionsfunktion für $L_1 \leq_p L_2$, und g die für $L_2 \leq_p L_3$. Dann wählen wir $h := g \circ f$ (also g angewendet auf f). Dies ist offensichtlich in polynomieller Zeit berechenbar, und es gilt

$$w \in L_1 \;\Leftrightarrow\; f(w) \in L_2 \;\Leftrightarrow\; g(f(w)) \in L_3 \;\Leftrightarrow\; h(w) \in L_3$$

(d) Sei $L \in \mathsf{coNP}$ beliebig. Es gilt also $\overline{L} \in \mathsf{NP}$. Wir wissen, dass SAT NP-vollständig ist, somit insbesondere NP-hart, woraus $\overline{L} \leq_p \mathsf{SAT}$ folgt. Durch Anwenden der (b) erhalten wir $L \leq_p \overline{\mathsf{SAT}}$, und aus der (a) und der (c) folgt nun $L \leq_p \mathsf{TAUTOLOGY}$.

Knobelaufgabe H12.4. (*Teilerfolg*)

Wir betrachten wieder das KRP aus Aufgabe H12.2. Sie haben leider festgestellt, dass es unmöglich ist, alle von Dr. Evilsparzas Lasern zu stoppen, wollen aber nicht aufgeben. Gibt es vielleicht eine Möglichkeit, zumindest die meisten Laser aufzuhalten?

(a) Zeigen Sie: Wenn die Flugbahn jeder Drohne mit höchstens einer anderen Drohne kollidieren würde, und jeder Laser von mindestens zwei Drohnen gestoppt werden kann, ist es möglich, mindestens $\frac{3}{4}$ der Laser zu stoppen.

(b) Beschreiben Sie ein Verfahren, das eine entsprechende Lösung in polynomieller Zeit auf einer DTM berechnet.

Hinweis: Vereinfachen Sie zunächst die Instanz und betrachten dann eine sinnvolle Teilmenge F der möglichen Lösungen. Wenn die Lösungen aus F im Durchschnitt mindestens $\frac{3}{4}$ der Laser stoppen, dann tut es die beste Lösung aus F auch.

Lösungsskizze. **(a)** Zunächst können wir das Problem vereinfachen, indem wir alle Drohnen, die mit keiner anderen Drohne kollidieren, aktivieren. Diese Drohnen und die von ihnen gestoppten Laser können wir nun entfernen, und erhalten eine Instanz, in der jede Drohne mit *genau* einer anderen kollidiert, und jeder Laser von mindestens zwei Drohnen gestoppt wird.

Nun können wir die Drohnen in Paare (d, b) aufteilen, die jeweils nur miteinander kollidieren. Wir betrachten im Folgenden nur Lösungen, bei denen genau einer dieser Drohnen fliegt. Sei also $V := S \cap D{\times}D$ die Menge dieser Paare. Für jede Funktion $f : V \to \{0,1\}$ existiert nun eine Lösung $M_f := \{d_{f((d_0,d_1))} : (d_0, d_1) \in V\}$, indem wir für jedes Paar die Drohne entsprechend f wählen.

Wir für einen Laser $l \in L$ definieren wir $r(f,l)$ als 1, wenn die Lösung M_f den Laser l stoppt, und 0 sonst. Wir berechnen nun die durchschnittliche Anzahl σ an Laser, die von einer Lösung M_f gestoppt werden, also

$$\sigma := \frac{1}{2^{|V|}} \sum_{f:V\to\{0,1\}} \sum_{l\in L} r(f,l)$$

Für einen Laser $l \in L$ gilt nun, dass es höchstens $2^{|V|-2}$ Möglichkeiten gibt, f so zu wählen, dass l *nicht* gestoppt wird: Für l gibt es mindestens zwei Drohnen d, b. Falls d, b kollidieren würden, fliegt immer mindestens eine davon und l würde immer gestoppt werden. Ansonsten muss f sowohl für d als auch für b jeweils den Partner wählen. Insbesondere gibt es also immer $2^{|V|} - 2^{|V|-2} = \frac{3}{4} \cdot 2^{|V|}$ Möglichkeiten, f zu wählen, sodass l gestoppt wird. Nun gilt

$$\sigma = \frac{1}{2^{|V|}} \sum_{f:V\to\{0,1\}} \sum_{l\in L} r(f,l) = \frac{1}{2^{|V|}} \sum_{l\in L} \sum_{f:V\to\{0,1\}} r(f,l) \geq \frac{1}{2^{|V|}} \sum_{l\in L} \frac{3}{4} \cdot 2^{|V|} = \frac{3}{4} \cdot |L|$$

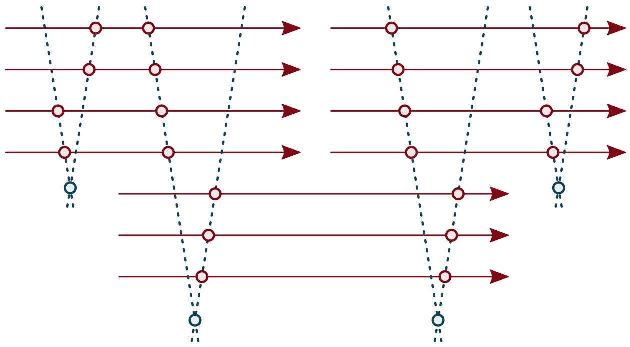
Im Durchschnitt über alle f werden also $\frac{3}{4}$ der Laser gestoppt, und, da das Maximum immer höher als der Durchschnitt ist, gibt es somit ein f, dass mindestens $\frac{3}{4}$ der Laser stoppt.

(b) Wir können das Verfahren aus der (a) erweitern. Zunächst vereinfachen wir die Instanz wie in (a), danach entfernen wir zusätzlich jeden Laser, der von zwei Drohnen, die kollidieren, gestoppt werden kann. (Diese Laser werden immer gestoppt.) Nun wissen wir, dass es für jeden Laser l, der von k Drohnen gestoppt werden kann, genau $2^{|V|-k}$ mögliche Wahlen von f gibt, sodass l gestoppt wird. (Die k Drohnen dürfen jeweils nicht fliegen, und die anderen $|V| - k$ sind beliebig.) Sei also $N(l) := |\{\{l\}{\times}D{\cap}S\}|$ die Anzahl der Drohnen, die Laser l stoppen könnten. Nun gilt

$$\sigma \overset{\text{(a)}}{=} \frac{1}{2^{|V|}} \sum_{l\in L} \sum_{f:V\to\{0,1\}} r(f,l) = \frac{1}{2^{|V|}} \sum_{l\in L} 2^{|V|-N(l)} = \sum_{l\in L} 2^{-N(l)}$$

Wir können den Durchschnitt also in polynomieller Zeit ausrechnen. Nun wählen wir ein beliebiges Paar $(d, b) \in V$ und betrachten die beiden Instanzen, die wir erhalten, wenn wir p (bzw. b) aktivieren und b (bzw. p) löschen. Für diese beiden Instanzen berechnen wir die Durchschnitte σ_1, σ_2. (Dazu müssen wir die sicher gestoppten Laser zunächst entfernen, und danach wieder aufaddieren.) Es gilt $\sigma = (\sigma_1 + \sigma_2)/2$, also gibt es ein $i \in \{1, 2\}$ mit $\sigma_i \geq \sigma$. Wir wählen nun dieses i und arbeiten rekursiv auf dieser Instanz weiter. Da die durchschnittliche Anzahl an gestoppten Lasern nie sinkt, finden wir also nach $|V|$ Schritten eine Instanz, in der $\frac{3}{4}$ der Laser gestoppt wurden.

Greedy Algorithmen, die immer die Drohne wählen, die die meisten Laser stoppt, funktionieren nicht. Hier ist ein Gegenbeispiel:



Der Greedy Algorithmus würde nur die oberen 8 Laser abdecken. Allerdings ist $8 < \frac{3}{4} \cdot 11$, somit würde er also nicht genügend Laser stoppen.

Eine andere Variante wählt immer die Drohne, die am wenigsten Laser stoppt, und entfernt diese.