Prof. Javier Esparza
Philipp Czerner, Martin Helfrich

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2021 – Hausaufgabenblatt 12

- Diese Woche werden alle Aufgaben korrigiert.
- Wenn Sie einen Beweis aufstellen, von dem Sie wissen, das einzelne Schritte problematisch oder unvollständig sind, merken Sie dies bitte in Ihrer Lösung an, damit wir das bei der Korrektur positiv berücksichtigen können.
- 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 gültig 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).
- Sie dürfen verwenden, dass Addition, Subtraktion und Multiplikation binäre dargestellter Zahlen in polynomieller Zeit berechnet werden kann.
- $0 \in \mathbb{N}$

Aufgabe H12.1. (*Eine Tautologie ist eine Tautologie*) 1+1+1+1 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 $\mathsf{coNP} := \{\overline{L} : L \in \mathsf{NP}\}$. 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 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

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

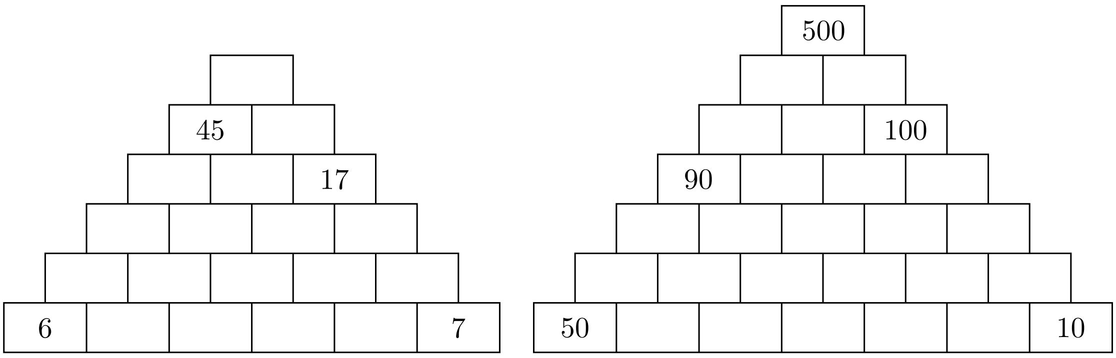
(c) Sei f die Reduktionsfunktion für $L_1 \leq_p L_2$, und g die für $L_2 \leq_p L_3$. Dann wählen wir $h := g \circ f$ (also g angewendet auf f). Dies ist offensichtlich in polynomieller Zeit berechenbar, und es gilt

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

(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 $L \leq_p \mathsf{TAUTOLOGY}$.

Aufgabe H12.2. (*NP = Number Pyramids?*) 2+1 Punkte

Der kleine Theo macht gerade seine Hausaufgaben (auch wenn eigentlich viel lieber den Pfützen beim Austrocknen zuschauen würde). Heute kamen Zahlenpyramiden dran:



Jedes Feld ist die Summe der beiden darunterliegenden Felder, und alle Felder sind mit ganzen Zahlen gefüllt (nur Pluszahlen). Formal ist das Problem $\mathsf{PYRAMID}$ wie folgt:

Eingabe: $w_1\$...\$w_k$, mit $w_i \in \{\square\} \cup bin(\mathbb{N})$ und $k = 1 + ... + n$ für ein $n \in \mathbb{N}$

Ausgabe: Existiert eine Zahlenpyramide, deren i-tes Feld w_i enthält, für $w_i \neq \square$?

Hierfür nummerieren wir die Felder durch, von oben nach unten und links nach rechts.

(a) Konstruieren Sie eine Zertifikatssprache für $\mathsf{PYRAMID}$ (nach Def. 6.7), sodass das Zertifikat für $w \in \mathsf{PYRAMID}$ höchstens Länge $p(|w|)$ hat, für ein Polynom p. Sie dürfen annehmen, dass jede lösbare Pyramide mit k Feldern eine Lösung hat, in der alle Zahlen höchstens $D \cdot 2^k$ sind, wobei D die größte vorgegebene Zahl ist.

Achtung: Eine Begründung, wieso die Zertifikate die vorgegebene Länge haben, ist erforderlich.

(b) Zeigen Sie, dass ein polynomiell beschränkter Verifikator für $\mathsf{PYRAMID}$ (bezüglich ihrer Zertifikate aus (a)) existiert, und somit $\mathsf{PYRAMID} \in \mathsf{NP}$.

Anmerkung: Können Sie die gezeigten Pyramiden (von Hand) lösen? Es gibt jeweils eine eindeutige Lösung. Für den geneigten Leser haben wir auch noch weitere Instanzen.

Lösungsskizze. **(a)** Sei $w \in \mathsf{PYRAMID}$ beliebig, mit $w = w_1\$...\$w_k$. Es gibt also eine Lösung der (partiellen) Zahlenpyramide w, und damit eine, deren größter Eintrag $D \cdot 2^k$ ist, wobei $D = \max\{(w_i)_2 : i \in \{1, ..., k\}\}$.

Unser Zertifikat enthält einfach die Binärdarstellungen der Zahlen, die in den Feldern der vollständig ausgefüllten Pyramide stehen. Wir wählen also $c := c_1\$...\$c_k$ mit $c_i := bin(x_i)$, wobei $x_i \in \mathbb{N}$ die Zahl ist, die in Feld i steht.

Wir wissen, dass $|w| \geq k$ und $|w| \geq \log_2 D$ gelten (letzteres, da die Binärdarstellung einer Zahl $x \in \mathbb{N}_{>0}$ genau $\lceil \log_2(x + 1)\rceil$ Stellen hat). Somit folgt

$$\begin{aligned}|c| &\leq k(1 + |c_1|) \leq k(2 + \log_2(x_1 + 1)) \leq k(3 + \log_2 x_1)\\ &\leq k(3 + \log_2 D + k) \leq |w|(3 + 2|w|)\end{aligned}$$

für $x_1 > 0$. Im Fall $x_1 = 0$ gilt stattdessen $|c| = 2k - 1 \leq 2|w| - 1 \leq |w|(3 + 2|w|)$. Für beide Fälle finden wir also ein polynomielle Schranke für $|c|$ in Abhängigkeit von $|w|$.

(b) Gegeben drei Binärwörter $u, v, w \in \{0, 1\}^*$ lässt sich in polynomieller Zeit überprüfen, ob $(u)_2 + (v)_2 = (w)_2$. Gegeben eine Eingabe $w\#c$ mit $w = w_1\$...\$w_k$ und $c = c_1\$...\$c_k$, müssen wir also überprüfen, dass

- $w_i = c_i$ für alle $w_i \neq \square$, und
- $(c_i)_2 + (c_j)_2 = (c_l)_2$ für alle Felder l, die über Feldern i, j liegen.

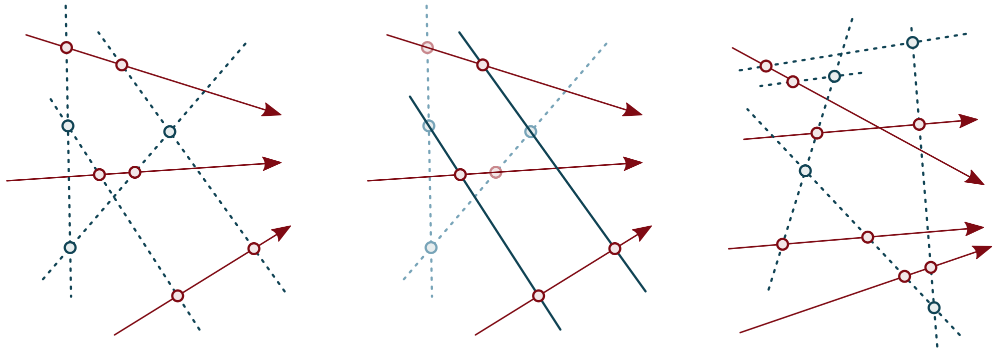
Beides lässt sich in polynomieller Zeit überprüfen, und muss jeweils höchstens k-mal ausgeführt werden. Da $|w| \geq k$, erhalten wir insgesamt einen polynomiell beschränkter Verifikator.

Aufgabe H12.3. (*Laser-focused*) 1+1+1+1+1 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 Geraden 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), eine Lösung dieser Instanz (2), und eine weitere, unlösbare Instanz (3). Die Laser sind jeweils als Pfeile gezeichnet und die möglichen Flugbahnen der Drohnen als gestrichelte Linien. Beachten Sie insbesondere, dass nicht jeder Schnittpunkt von zwei Graden bedeutet, dass sich die Drohnen bzw. Laser da kreuzen würden.

(a) Zeigen Sie KRP $\leq_p$ SAT.

(b) Konstruieren Sie ein „Gadget“, um eine boolesche Variable zu modellieren. Wir suchen eine Instanz $I_x = (L, D, S)$ mit $x, \overline{x} \in D$. Jede Lösung von I_x muss *entweder* x oder $\overline{x}$ enthalten, und es muss Lösungen S_1, S_2 mit $x \in S_1$, $\overline{x} \in S_2$ geben.

In der Vorlesung werden wir das NP-vollständige Problem 3KNF-SAT kennen lernen. Dies führen wir hier vorab schon einmal ein, da es als Einschränkung von SAT die Reduktion leichter macht.

Sei X eine Menge an Variablen. Die Menge der *Literale* ist $L := X \cup \{\neg x : x \in X\}$, enthält also alle Variablen und Negationen von Variablen. Eine *k-Klausel* ist eine Formel der Form $l_1 \vee l_2 \vee ... \vee l_k$ mit $l_1, ..., l_k \in L$, und eine Formel F ist in *k-konjunktiver Normalform (k-KNF)*, wenn $F = F_1 \wedge F_2 \wedge ... \wedge F_n$, wobei $F_1, ..., F_n$ jeweils k-Klauseln sind. Das Problem 3KNF-SAT ist nun

Eingabe: Aussagenlogische Formel F in 3-KNF

Ausgabe: Ist F erfüllbar?

Ein Beispiel für eine Instanz wäre also $(\neg x \vee y \vee z) \wedge (x \vee \neg y \vee z) \wedge (x \vee y \vee \neg z)$.

(c) Beschreiben Sie, wie Sie Klauseln der Länge 3 mithilfe ihres Gadgets aus (b) modellieren können, sodass jede Lösung der Instanz einer erfüllenden Belegung der Klausel entspricht, und umgekehrt. Demonstrieren sie Ihre Konstruktion anhand des Beispiels $x \vee \neg y \vee z$.

(d) Zeigen Sie 3KNF-SAT $\leq_p$ KRP.

(e) Konstruieren Sie eine KRP Instanz, die genau dann lösbar ist, wenn $(x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z)$ erfüllbar ist. Verwenden Sie hierzu Ihre Konstruktion aus (d).

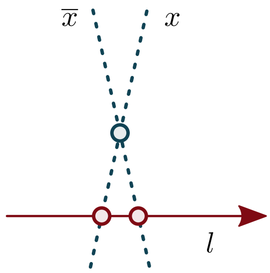
Lösungsskizze. **(a)** Sein (L, D, S) eine KRP-Instanz. Wir konstruieren nun eine SAT-

Instanz, also eine aussagenlogische Formel, die genau dann erfüllbar ist, wenn (L, D, S) eine Lösung hat. Wir verwenden eine Variable x_d für jede Drohne $d \in D$. Unsere Formel F ist nun

$$\Big(\bigwedge_{l\in L}\ \bigvee_{(l,d)\in S} x_d\Big) \wedge \Big(\bigwedge_{(d,b)\in S\cap D^2} (\neg x_d \vee \neg x_b)\Big)$$

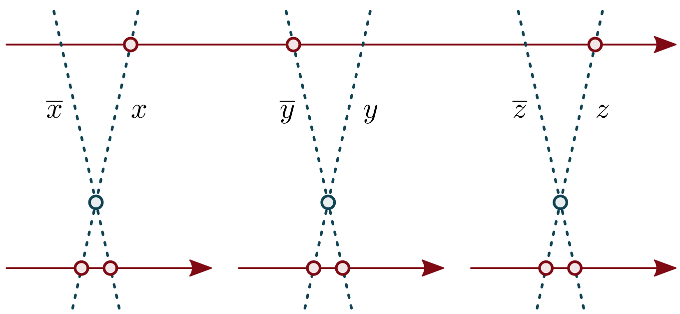
Anschaulich gesprochen fordern wir also für jeden Laser, der von Drohnen $d_1, ..., d_k$ geschnitten wird, dass $x_{d_1} \vee ... \vee x_{d_k}$ (jeder Laser muss zerstört werden), und für jedes Paar Drohnen (d, b), deren Flugbahnen sich überschneiden, dass $\neg x_d \vee \neg x_b$ (keine zwei Drohnen kreuzen sich). Aus der Konstruktion ist bereits offensichtlich, dass die Reduktion korrekt ist.

(b) Unsere Konstruktion sieht folgendermaßen aus:



Formal verwenden wir also zwei Drohnen $D := \{x, \overline{x}\}$ und einen Laser $L := \{l\}$. Die Schnittpunkte sind $S := \{(x, \overline{x}), (l, x), (l, \overline{x})\}$.

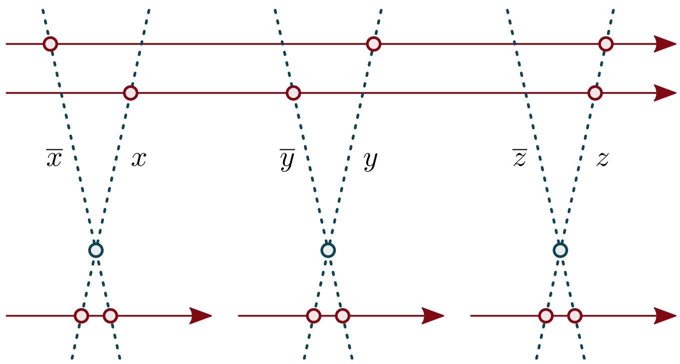
(c) Wir fügen eine Kopie von I_v für jede Variable $v \in \{x, y, z\}$ ein. Für dir Klausel erstellen wir dann einen Laser, und Schnittpunkte mit genau den Literalen aus der Klausel. Für das Beispiel $x \vee \neg y \vee z$ erhalten wir folgendes.



(d) Hier müssen wir lediglich ein Gadget I_v für jede Variable erstellen, und dann für jede Klausel einen Laser wie in (c) einfügen. Dies lässt sich offensichtlich in polynomieller Zeit machen. Wir zeigen nun „KRP lösbar $\Leftrightarrow$ 3KNF-SAT lösbar“:

- „$\Rightarrow$“ Sei M die Lösung des KRP. Für jede Variable v ist nach Konstruktion von I_v entweder $v \in M$ oder $\overline{v} \in M$. Wir wählen nun eine entsprechende Belegung der 3KNF-SAT Instanz. Unter dieser Belegung muss nun jede Klausel erfüllt sein, da wir für jede Klausel einen entsprechenden Laser erstellt haben.
- „$\Leftarrow$“ Sei σ eine erfüllende Belegung der 3KNF-SAT Formel. Wir wählen nun $M := \{v : \sigma(v) = 1\} \cup \{\overline{v} : \sigma(v) = 0\}$. Da für jede Variable v die Drohnen v und $\overline{v}$ nicht gleichzeitig fliegen, kollidieren keine Drohnen. Jeder Laser in I_v wird zerstört, da v oder $\overline{v}$ immer fliegt, und die Laser für die Klauseln werden auch zerstört, da σ erfüllend ist.

(e) Die Instanz sieht folgendermaßen aus:



Bonusaufgabe H12.4. (*Teilerfolg*) 1+1 Punkte

Achtung: Die Bonusaufgabe wird diese Woche separat korrigiert. Bitte fertigen Sie für diese Aufgabe eine getrennte Lösung an.

Wir betrachten wieder das KRP aus Aufgabe H12.3. 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

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

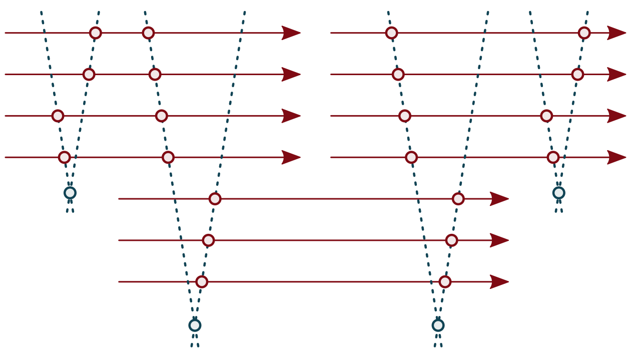
Im Durchschnitt über alle f werden also $\frac{3}{4}$ der Laser gestoppt, und, da das Maximum immer höher als der Durchschnitt ist, gibt es somit ein f, dass mindestens $\frac{3}{4}$ der Laser stoppt.

(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.