Aufgabe 3: Berechnen Sie, wie viele Variable und Klausen die SAT-Codierung aus der Vorlesung benötigt, um die Eigenschaft $W(2,k) > n$ zu prüfen.

- Variablen:

Es wird für jede Zahl $i \in \{1, \dots, n\}$ eine boolsche Variable x_i eingeführt (Mit $k = 2$) → Somit genau n Variablen ($x_1, x_2 \dots, x_n$)

Mit $\begin{cases} x_i = 0 \Leftrightarrow x_i \text{ Farbe } 1 \\ x_i = 1 \Leftrightarrow x_i \text{ Farbe } 2 \end{cases}$

- Klausen:

+ Für arithmetische Progression der Form:

$x_i, \quad x_{i+d}, x_{i+2d}, \dots, \quad x_{i+(k-1)d}$

Mit $0 < i, i+d \to i+(k-1)d \leq n$

→ Für jede Progression müssen 2 Klausen eingeführt werden um zu verhindern, dass alle Elemente gleich 0 ~~Null~~ oder 1 sind

↳ $(x_i \vee x_{i+d} \vee x_{i+2d} \vee \dots \vee x_{i+(k-1)d})$

und $(\overline{x_i} \vee \overline{x_{i+d}} \vee \overline{x_{i+2d}} \vee \dots \vee \overline{x}_{i+(k-1)d})$

→ Anzahl Klausen = 2. Gesamtzahl der arithmetische Progressionen

⊕ Betrachtung Anzahl arithmetische Progression

Größe k, Max n

→ $i + (k-1)d \leq n$

⊕ $d_{min} = 1$, d_{max}: $d \leq \frac{n-i}{k-1} < \frac{n-1}{k-1}$ $(i \geq 1)$

→ $d_{max} = \left\lfloor \frac{n-1}{k-1} \right\rfloor$

Für jede d die Anzahl von arithmetischen Progressionen ist $\begin{cases} i + (k-1)d : i = \cancel{0} \\ i = 1 \end{cases}$ → $i + (k-1)d = n$

(Start von dem letzten Element)

→ $i = n - (k-1)d$ ist max)

(Dem letzten Element ~~ist~~ ~~text~~

→ insgesamt $n - (k-1)d$ (arit.. Progression)

→ Gesamte Anzahl von arithmetischen Progressionen

$$\sum_{d=1}^{\lfloor \frac{n-1}{k-1} \rfloor} n-(k-1)d$$

Conclusion: Nr. Variablen: n

Nr. Klausen: $2 \sum_{d=1}^{\lfloor \frac{n-1}{k-1} \rfloor} n-(k-1)d$

Aufgabe 4: Pythagoräische Triple:

- Abschätzung der Variable und Klauseln
  + Für Zahlen 1 bis n haben wir genau n Variablen
  + Klausel: Für jede (a, b, c) gibt es 2 Klausel
    $(x_a \vee x_b \vee x_c) \wedge (\overline{x_a} \vee \overline{x_b} \vee \overline{x_c})$

Daher: Anzahl Klausel $= 2 \cdot$ Anzahl pytagoräische Tripel
$T(n)$

Wir können $T(n)$ in 2 Schritten abschätzen.

Zunächst parametrisieren wir alle pyta. Tripel und drücken dann die Anzahl $T(n)$ als Summe über die Parameter aus, die wir anschließend asymptotisch abschätzen

1. Parametisierung:

→ Jede Tripel a, b, c lässt sich durch (Euclid's Formular)

$a = k(m^2-n^2)$, $b = k \cdot (2mn)$, $c = k(m^2+n^2)$

Mit $m > n > 0$, $m, n \in \overline{\mathbb{N}}$, $\gcd(m,n) = 1$, $m \not\equiv n \mod 2$

Für jedes gegebenes Tripel is der Faktor k durch Bedingung $c \leq n$ beschränkt (c is die größte Zahl im Tripel)

$\rightarrow k(m^2+n^2) \leq n_{max} \iff k \leq \frac{n_{max}}{m^2+n^2}$

Somit trägt jedes primitives Tripel zu höchsten $\left\lfloor \frac{n_{max}}{m^2+n^2} \right\rfloor$ Tripel bei:

$$\Rightarrow T(n) \approx \sum_{\substack{m>n>0 \\ m,n \text{ primitiv}}} \left\lfloor \frac{n_{max}}{m^2+n^2} \right\rfloor \approx n_{max} \sum_{\substack{\sqrt{\frac{n_{max}}{2}} > m > n > 0 \\ m,n \text{ prim}}} \frac{1}{m^2+n^2}$$

wächst wie $\log(n)$

2. Abschätzung

Wir betrachten

$$S = \sum_{m,n \geq 1} \frac{1}{m^2+n^2}$$

→ Zieht $\frac{m}{n} = \cot\varphi = \frac{\sin\varphi}{\cos\varphi}$

$m = r\cos\varphi \quad , \quad n = r \cancel{\cos\varphi} \sin\varphi$

Beim Herleitung über den Jacobian Matrix

$$J = \begin{pmatrix} \frac{d(r\cos\varphi)}{dr} & \frac{d(r\cos\varphi)}{d\varphi} \\ \frac{d(r\sin\varphi)}{dr} & \frac{\frac{dr\sin\varphi}{}}{d\varphi} \end{pmatrix} = \begin{pmatrix} \cos\varphi & -r\sin\varphi \\ \sin\varphi & r\cos\varphi \end{pmatrix}$$

$$\det J = r\cos^2\varphi + r\sin^2\varphi = r$$

$$\rightarrow dm\,dn = r\,dr\,d\varphi$$

$$\rightarrow S = \int_{r=r_0}^{R} \int_{\varphi=0}^{2\pi} \frac{1}{r^2} \cdot r \cdot dr\,d\varphi \qquad (R \simeq n_{max})$$

$$= \int_{r=r_0}^{R} \int_{\varphi=0}^{2\pi} \frac{1}{r}\,dr\,d\varphi = \int_{r=r_0}^{R} \frac{1}{r}\,dr \cdot \left(\int_{\varphi=0}^{2\pi} d\varphi \right)$$

$$= 2\pi \cdot \int_{r=r_0}^{R} \frac{1}{r}\,dr = 2\pi\left(\ln(R) - \ln(r_0)\right)$$

skaliert $\Rightarrow O(\log n)$

$\Rightarrow S \sim O(\ln n)$

→ $T(n)$ wächst wie $O(n \log n)$