Satz Gegeben 2 deterministische kontextfreie Sprachen L_1 und L_2. Dann ist es unentscheidbar, ob:

(a) $L_1 \cap L_2 = \emptyset$?

(b) $|L_1 \cap L_2| = \infty$?

(c) $L_1 \cap L_2$ kf ?

(d) $L_1 \subseteq L_2$?

Bew: Da die im letzten Beweis konstruierte Spr. det. kf. waren, gelten die Ergebnisse auch für diese.

Satz Gegeben eine kf. Gramm. G, dann ist es unentscheidbar, ob

(a) G mehrdeutig ist,

(b) $L(G)$ kontextfrei ist,

(c) $L(G)$ regulär ist, und

(d) $L(G)$ det. kf ist.

Bew:

(a) Gegeben G_1 und G_2 aus dem letzten Beweis (Reduktion von PCP). Sei G_3 die Grammatik mit $L(G_3) = L(G_1) \cup L(G_2)$. Diese ist effektiv konstruierbar.

Das PCP hat eine Lösung gdw. $L(G_1) \cap L(G_2) \neq 0$.

D.h. das PCP hat eine Lösung gdw. wenn es ein Wort w gibt mit $w \in L(G_1)$ und $w \in L(G_2)$. Dann wäre G_3 mehrdeutig. Ansonsten ist es so, dass beide Gramm. det. kf. sind, d.h. auch eindeutig. D.h. das PCP hat eine Lösung gdw. G_3 ist mehrdeutig.

$K = ((x_1, y_1), \ldots, (x_k, y_k))$

$((01, 0), (1, 10))$ $\quad x = 011$, $y = 010$ ✗ $\quad \cancel{101}$, $\cancel{100}$

(b) Seien G_1' und G_2' wieder wie im letzten Bew.
$L(G_1') = \overline{L(G_1)}$ und $L(G_2') = \overline{L(G_2)}$ (konstruierbar, da G_1 und G_2 det kf). Sei G_4 die Gramm. mit

$$L(G_4) = L(G_1') \cup L(G_2').$$

Dann hat das PCP eine Lösung gdw. $\underline{L(G_1) \cap L(G_2)} =$

$$\overline{\overline{L(G_1) \cap L(G_2)}} = \overline{\overline{L(G_1)} \cup \overline{L(G_2)}} = \overline{L(G_1') \cup L(G_2')} = \overline{L(G_4)} \neq \emptyset$$

gdw. $\overline{L(G_4)}$ nicht kf.

(c+d) Wenn das PCP keine Lösung hat, dann gilt
$\underline{L(G_4) = \Sigma^*}$. Wenn das PCP eine Lösung hat, dann gilt
$L(G_4) = \Sigma^* - (\underline{L(G_1) \cap L(G_2)})$. D.h. dann ist
$L(G_4)$ weder regulär noch det. kf., da dies ja sonst auch für $\overline{L(G_4)}$ gelten müsste.

□

Satz Gegeben eine kf Sprache L_1 und eine reg. Sprache L_2, so ist unentscheidbar ob $L_1 = L_2$.

Bew: Folgt aus dem letzte Bew für $L_2 = \Sigma^*$.

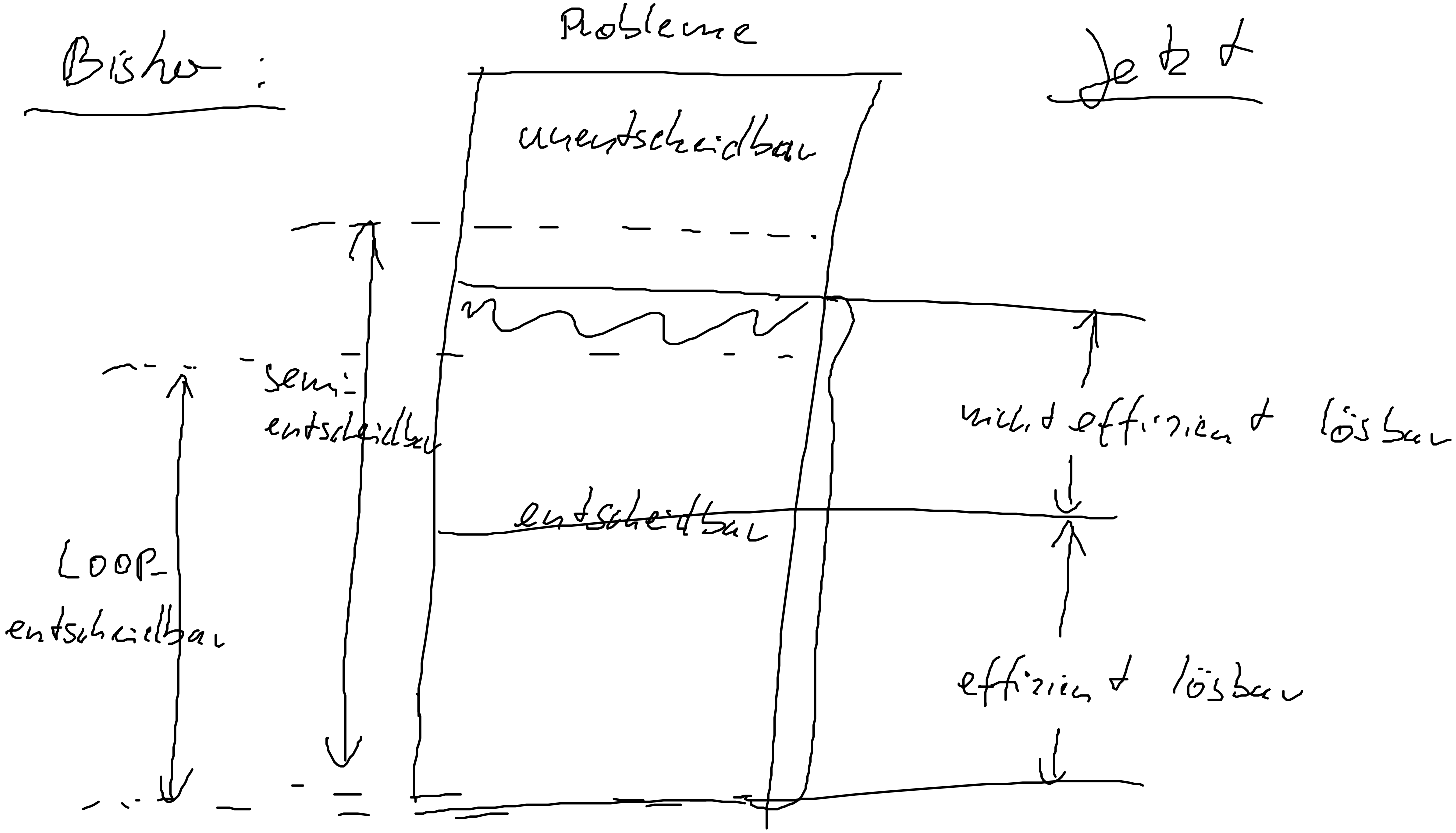
Satz Das Leerheitsproblem und das Endlichkeitsproble für Typ 1-Sprachen ist unentscheidbar.

Bew: Wir reduziere das Schnittproblem für kf. Sprachen auf das Leerheitsproblem für Typ 1-Spr. Da Typ 1-Spr. effektiv unter Schnitt abgeschlossen sind, ex eine Typ 1-Gram. G für gegebene kf. Gram G_1 und G_2 mit $L(G) = L(G_1) \cap L(G_2)$.

$\square$

3. Komplexitätstheorie

Welche Berechnungsressourcen werden benötigt, um ein gegebenes Problem zu lösen?



Effizient lösbar?

→ Laufzeit, die durch ein Polynom abgeschätzt werden kann, könnte auch $O(n^{1000})$, tatsächlich sind in der Praxis die meiste Probleme mit kleinen Exponenten lösbar

Wortproblem für kf. Sprache $O(n^3)$ (CYK-Alg.)

Nicht effizient lösbar

→ <u>superpolynomiale untere Schranke</u> $O(2^n)$ $O(n^{\log n})$

Warum überhaupt poly. Laufzeit?

→ Sehr robust über verschieden Maschinenmodelle.

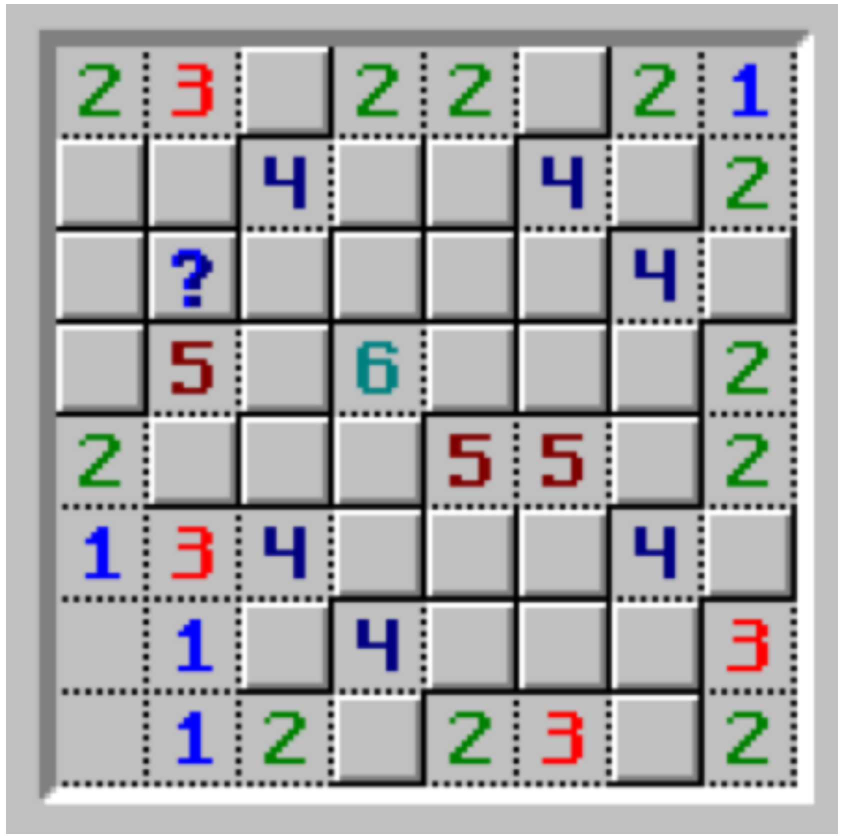
z.B. Mehrband-TMs können auf 1-Band-TMs simuliert werden: $O(n^2)$ Zeit.

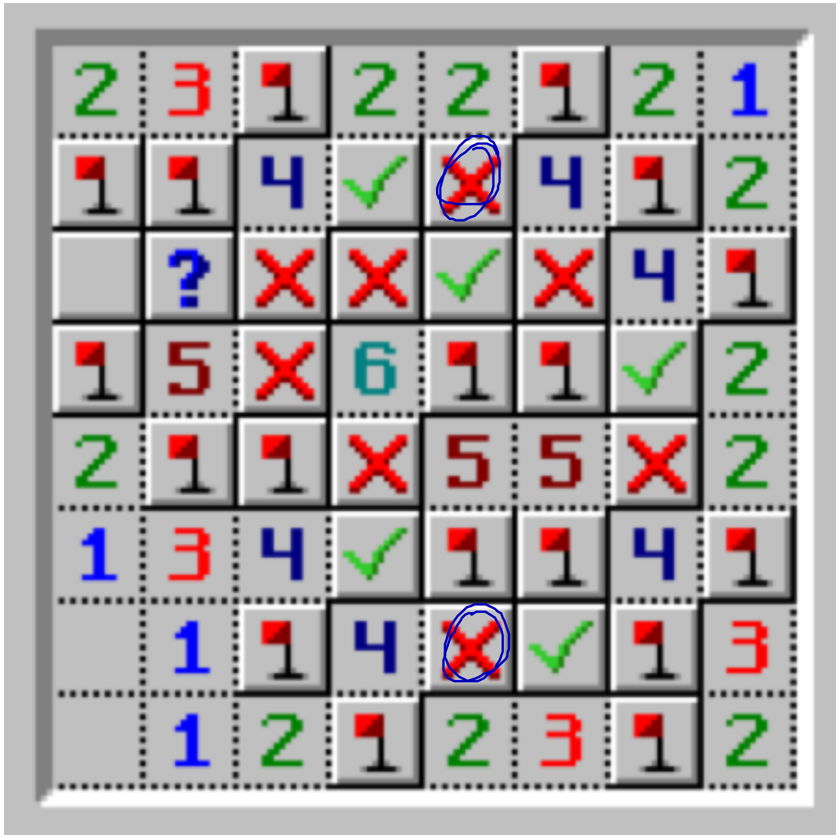
<u>Beobachtung</u>: Oft lassen sich keine poly. Entscheidungsverfahren finden, aber es lassen auch keine superpoly. unteren Schranken beweisen.

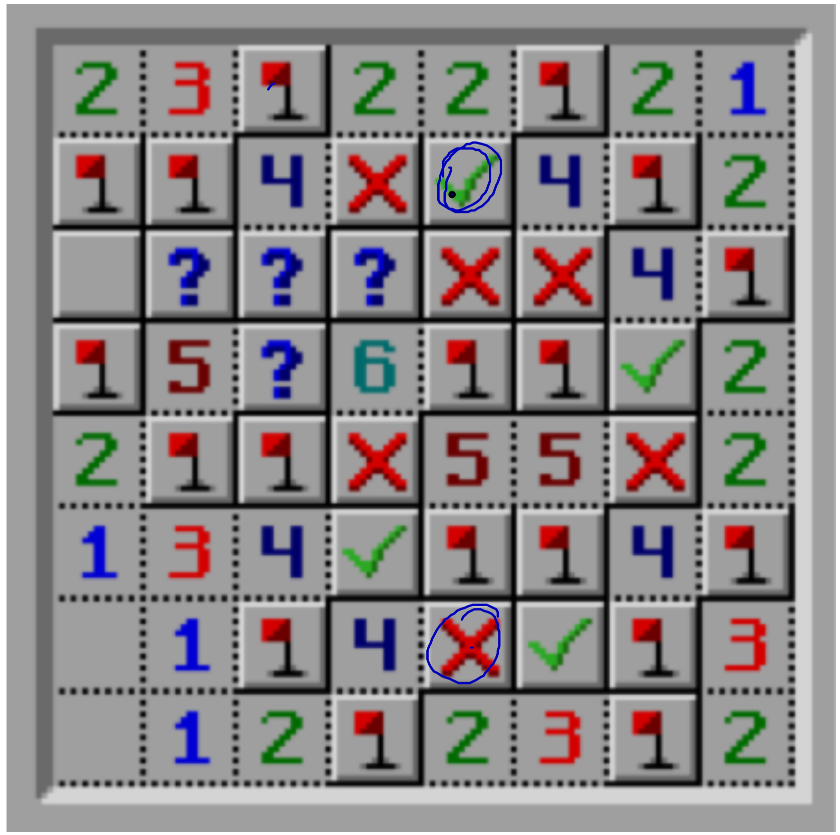
Bsp: Entscheidungsprobleme in Solitär-Spielen, wie z.B. Free Cell, Mine Sweeper ...

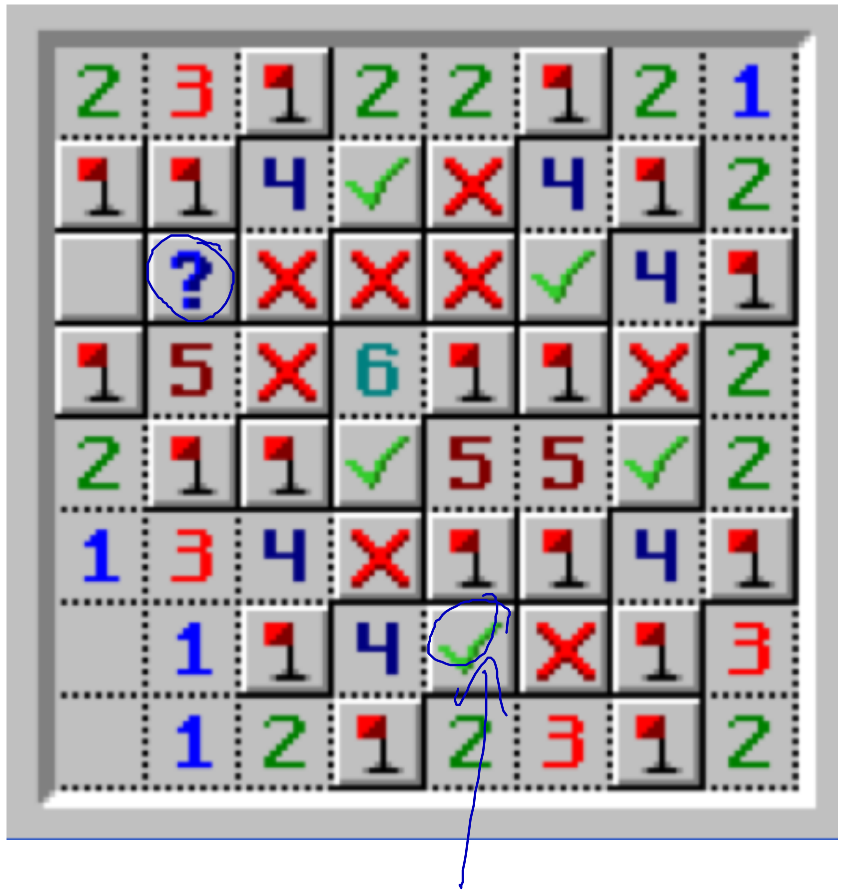
→ Wie bestimmt man einen guten Zug

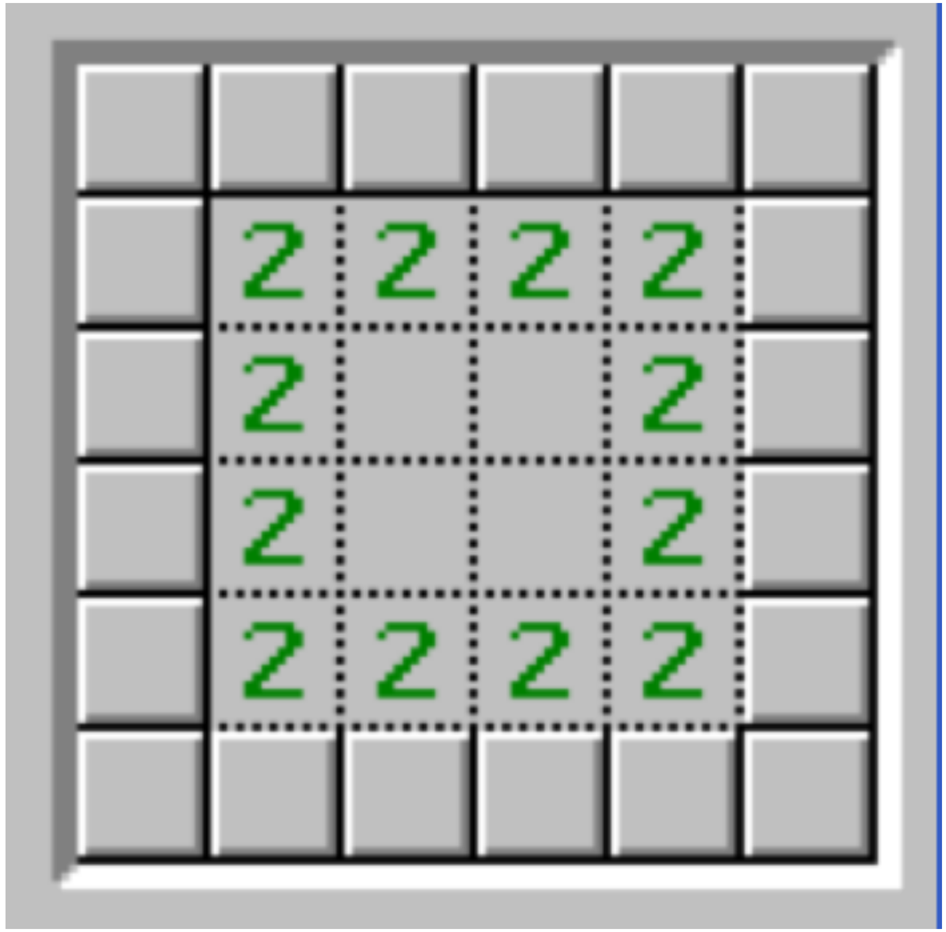
→ Wähle Feld, auf dem keine Mine liegen kann.

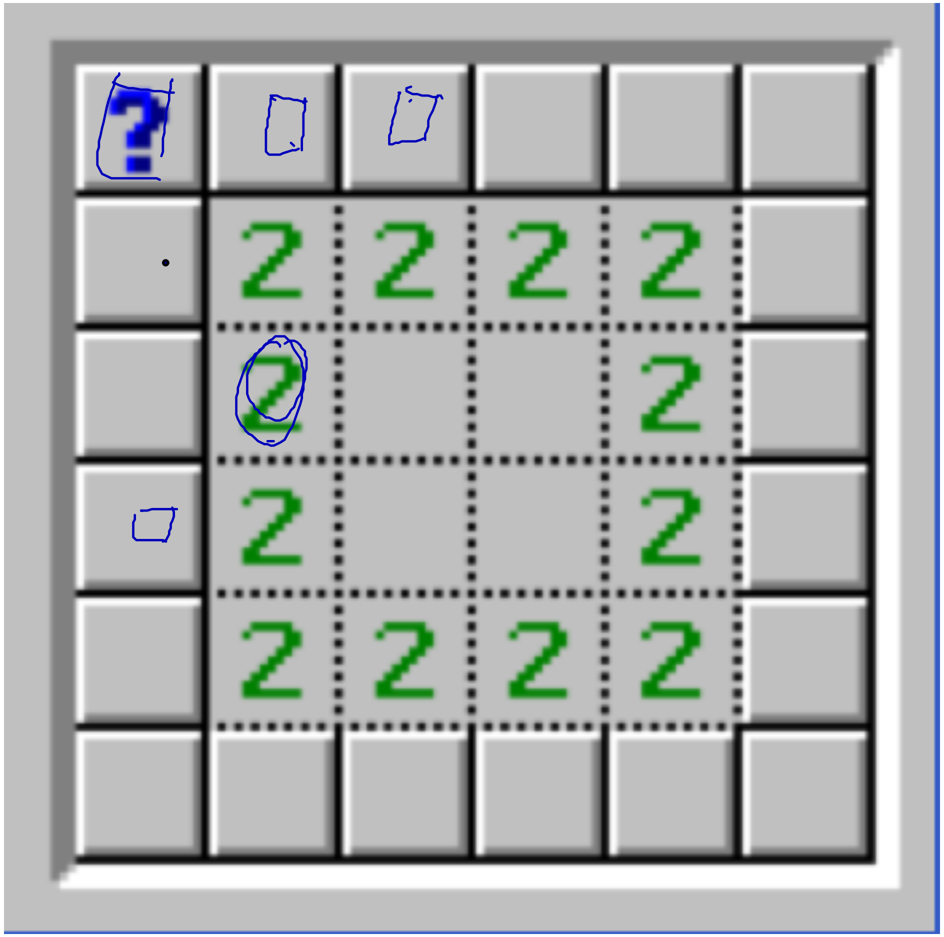

2 3 2 2 2 1
4 4 2
? 4
5 6 2
2 5 5 2
1 3 4 4
1 4 3
1 2 2 3 2

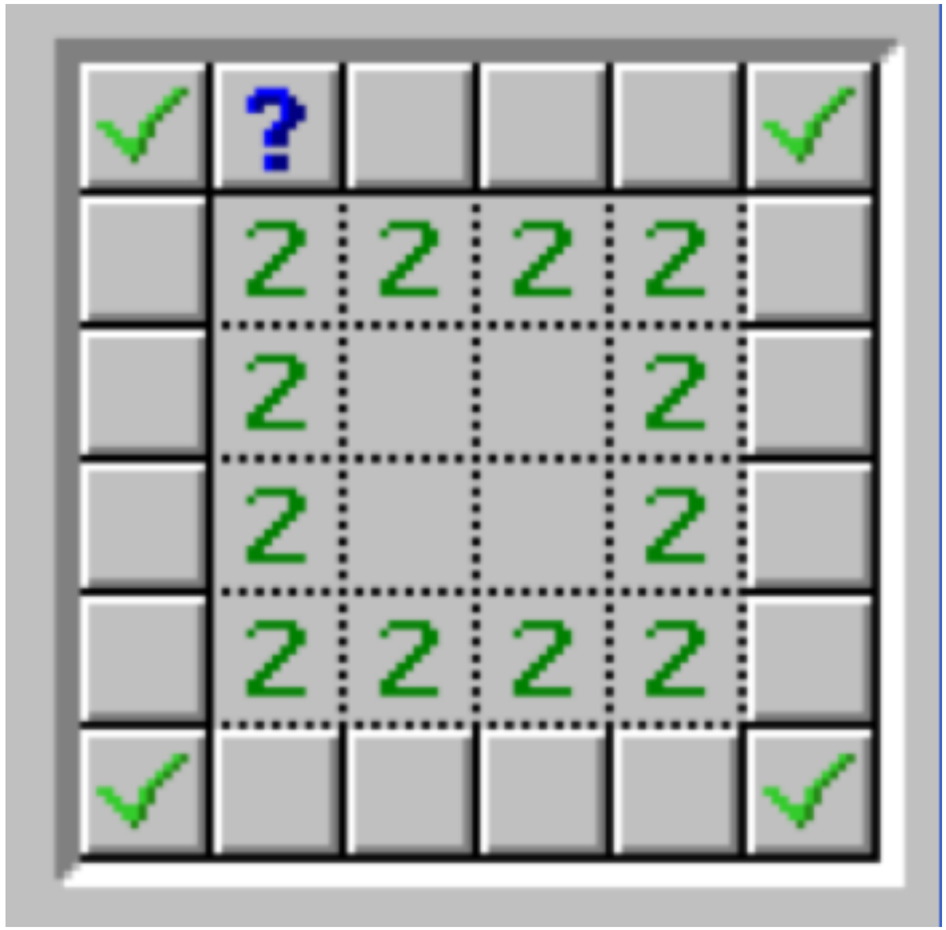


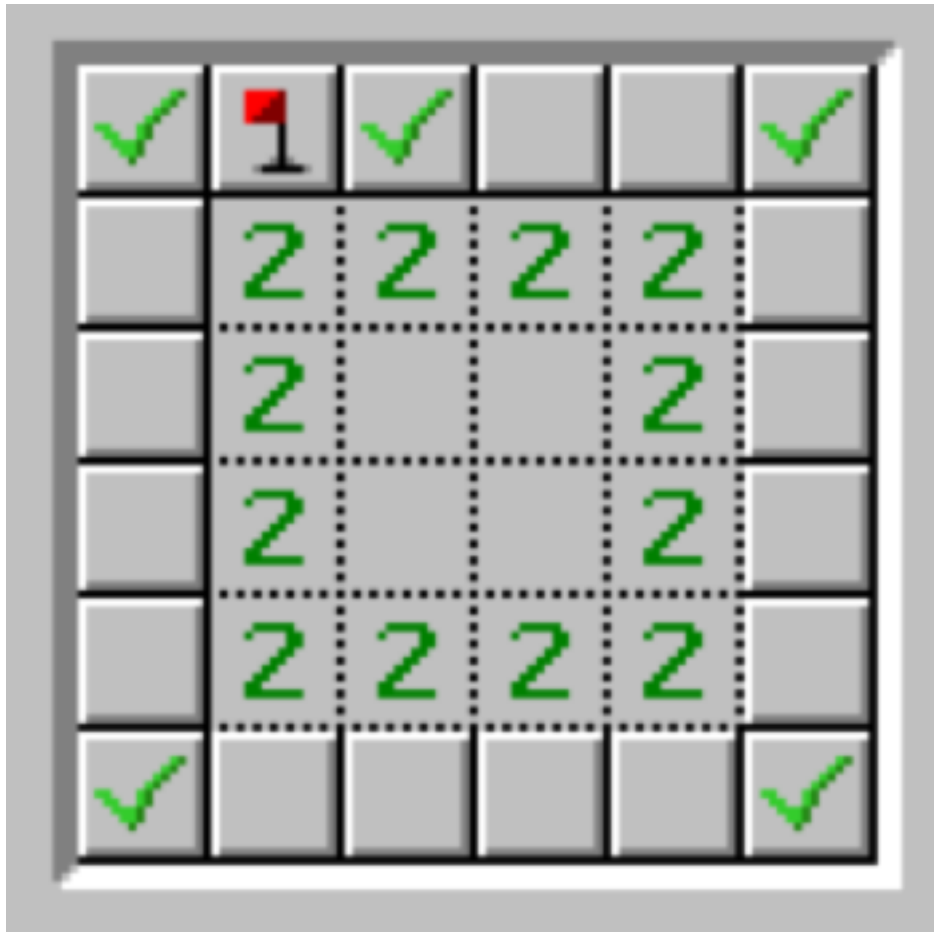


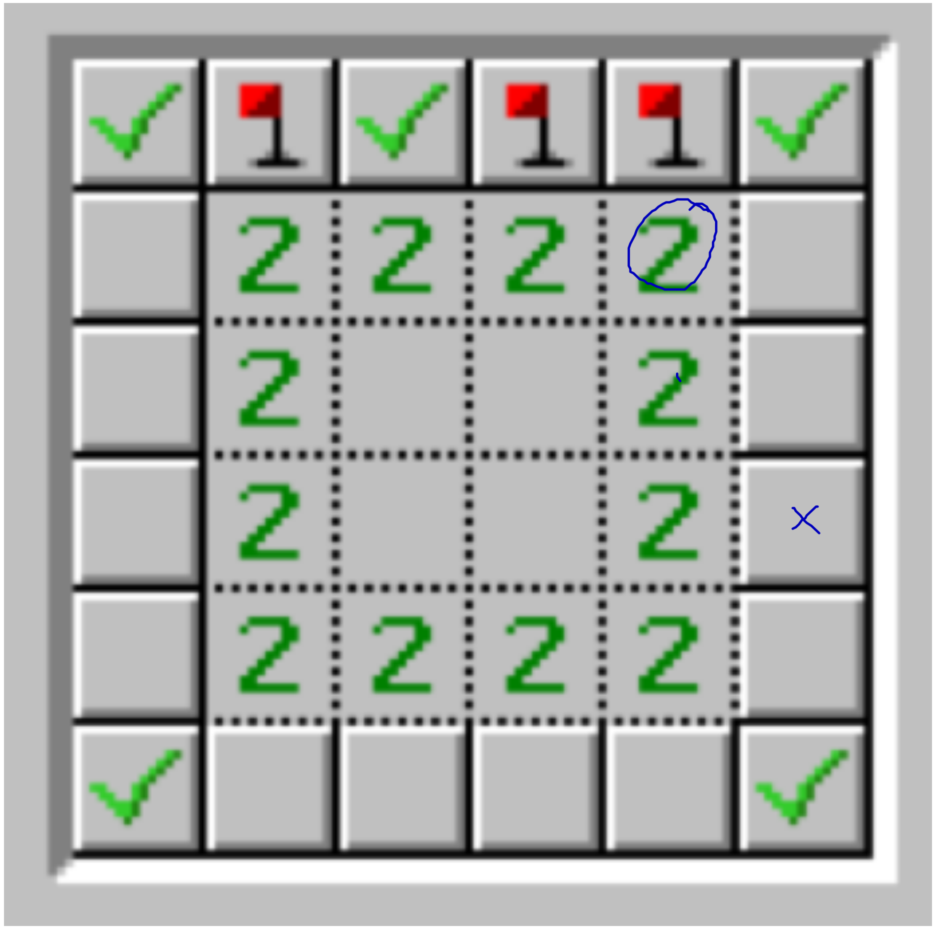


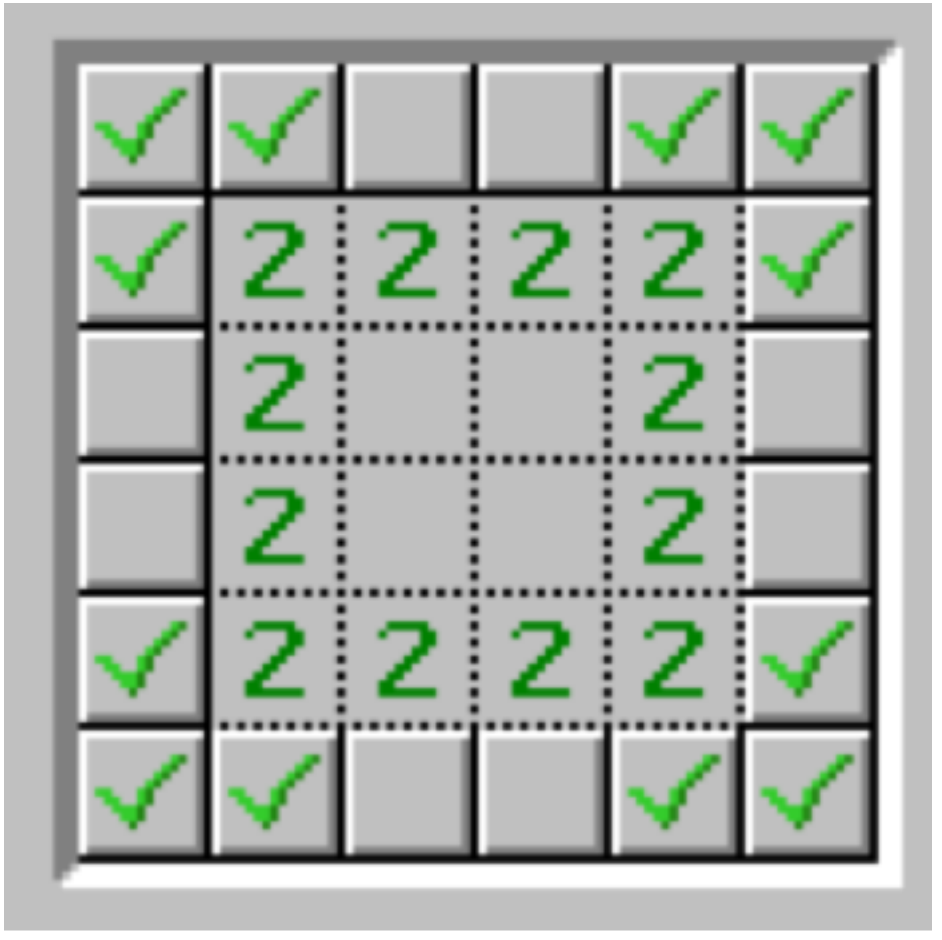

2 2 2 2
2 2
2 2
2 2 2 2

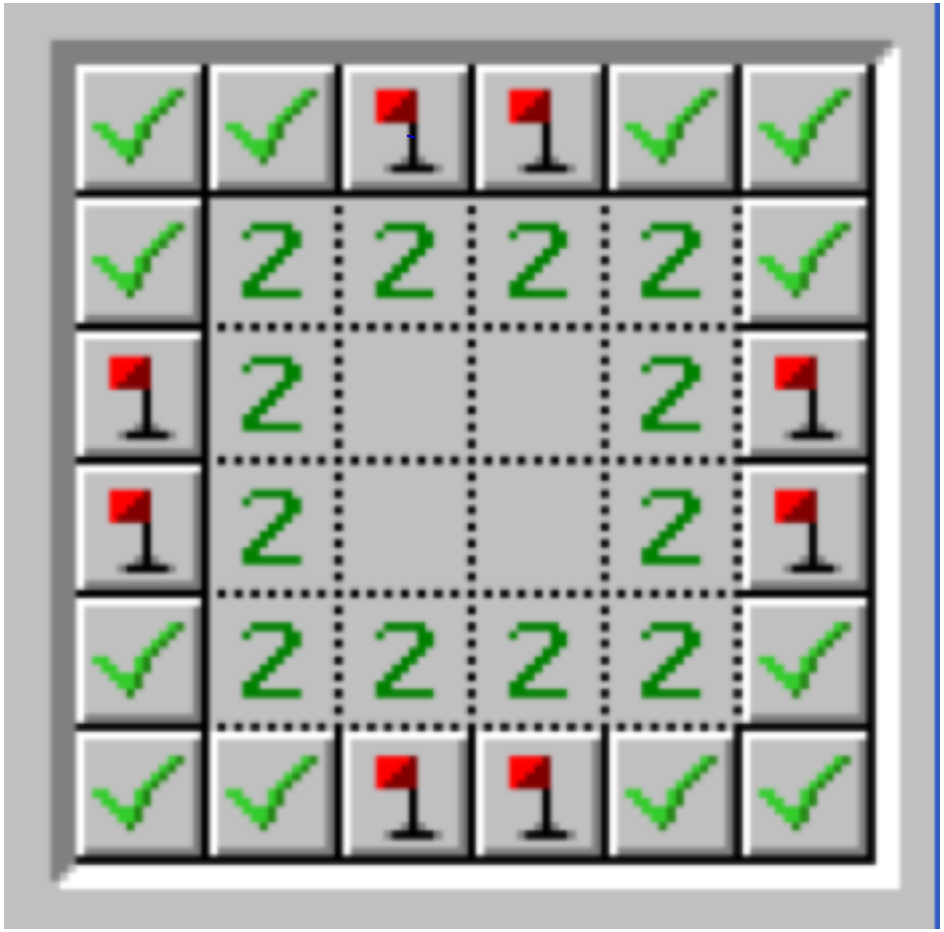

?
2
2
2
2
2
2
2
2
2
2
2
2


?
2 2 2 2
2 2
2 2
2 2 2 2


2 2 2 2
2 2
2 2
2 2 2 2







Beobachtung:

1) Es ist schwierig, zu entscheiden, ob ein bestimmtes Feld eine Mine enthalten könnte.

2) Sind alle Minen und alle Beobachtungen bekannt, dann ist es einfach, zu überprüfen, ob dies eine legale Belegung ist.

Um 1 zu lösen: Erzeuge vollständiges Szenario, das mit der aktuellen Situation übereinstimmt, und schaue ob dort eine Mine liegt. Um festzustellen, dass dort keine Minen liegen kann: Lauf über alle möglichen Szenarien.

→ Besser geht es (vermutlich) nicht!

da das Problem NP-vollständig ist

3.1 Komplexitätsklassen

<u>Def.</u> $time_M : \Sigma^* \to \mathbb{N}$ ist eine Fkt, die die Anzahl der Rechenschritte einer det. TM M auf der Eingabe $x \in \Sigma^*$ angibt.

<u>Def.</u> Sei $f : \mathbb{N} \to \mathbb{N}$ eine Fkt. Die Klasse $TIME(f(n))$ enthält alle Sprachen/Probleme A, für die es eine det. Mehrband-TM M mit $A = T(M)$ und $time_M(x) \leq f(x)$ gibt.

<u>Def.</u> Ein Polynom ist eine Fkt $p : \mathbb{N} \to \mathbb{N}$ der Form $p(n) = a_k \cdot n^k + a_{k-1} \cdot n^{(k-1)} + \ldots a_1 \cdot n + a_0$

$a_i \in \mathbb{N}$, $k \in \mathbb{N}$.

Def Die Komplexitätsklasse P ist definiert wie folgt:

$$P = \bigcup_{p \text{ Polynom}} TIME(p(n)) .$$

Bem: Alg. mit Laufzeit $n \cdot \log n$ sind auch polynomial, da abschätzbar n^2.

$\rightarrow$ P wird als Klasse der effizient lösbaren Probleme angesehen (auf sequentiellen Maschinen).