

NP vs. co-NP

SAT ist NP, da wir eine "Lösung" (oder Zertifikat) raten können ($\exists$ Belegung der Booleschen Var.) und dann können wir in poly. verifizieren, dass die Belegung die Formel wahr macht.

$$\underline{\text{UNSAT}} = \{ F \mid F \text{ ist unerfüllbar} \}$$

UNSAT $\in$ NP?

→ Was soll hier gezeigt werden, dass nur polynomial groß ist?

$$\text{co-NP} = \{ L \mid \bar{L} \in \text{NP} \}$$

Dabei wird das Komplement nicht bezüglich Σ^* gebildet, sondern bezüglich der Instanzensprache.

Bsp.: Bei SAT und UNSAT: Menge aller Formeln.

Vermutlich $\text{co-NP} \neq \text{NP}$, aber genauso oft wie
 $P \neq \text{NP}$.

Mine-Sweeping Problem (MSP)

Gegeben: Ein partiell aufgedecktes Mine-Sweeping-Feld mit Minenmarkierungen.

Gefragt: Gibt es ein "dazu passendes" vollständig aufgedecktes Feld?

Beweis:

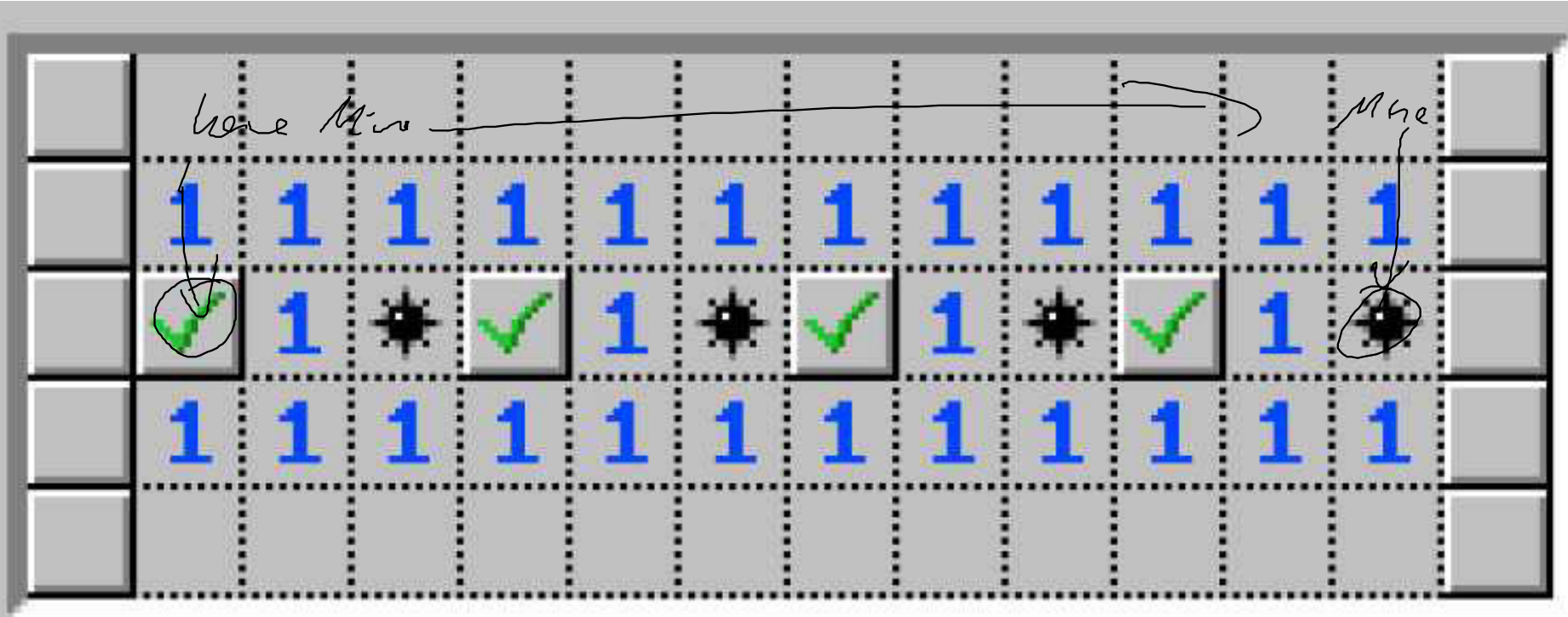
$\in NP$: Rate vollständig aufgedecktes Feld und überprüfe, ob das Feld "passend" ist.

NP -Härte: Reduktion $SAT \leq_p MSP$

Gebe eine Formel F , erzeuge dazu eine entsprechende MSP -Instanz I (ein partiell aufgedecktes Feld).

"Leitung"

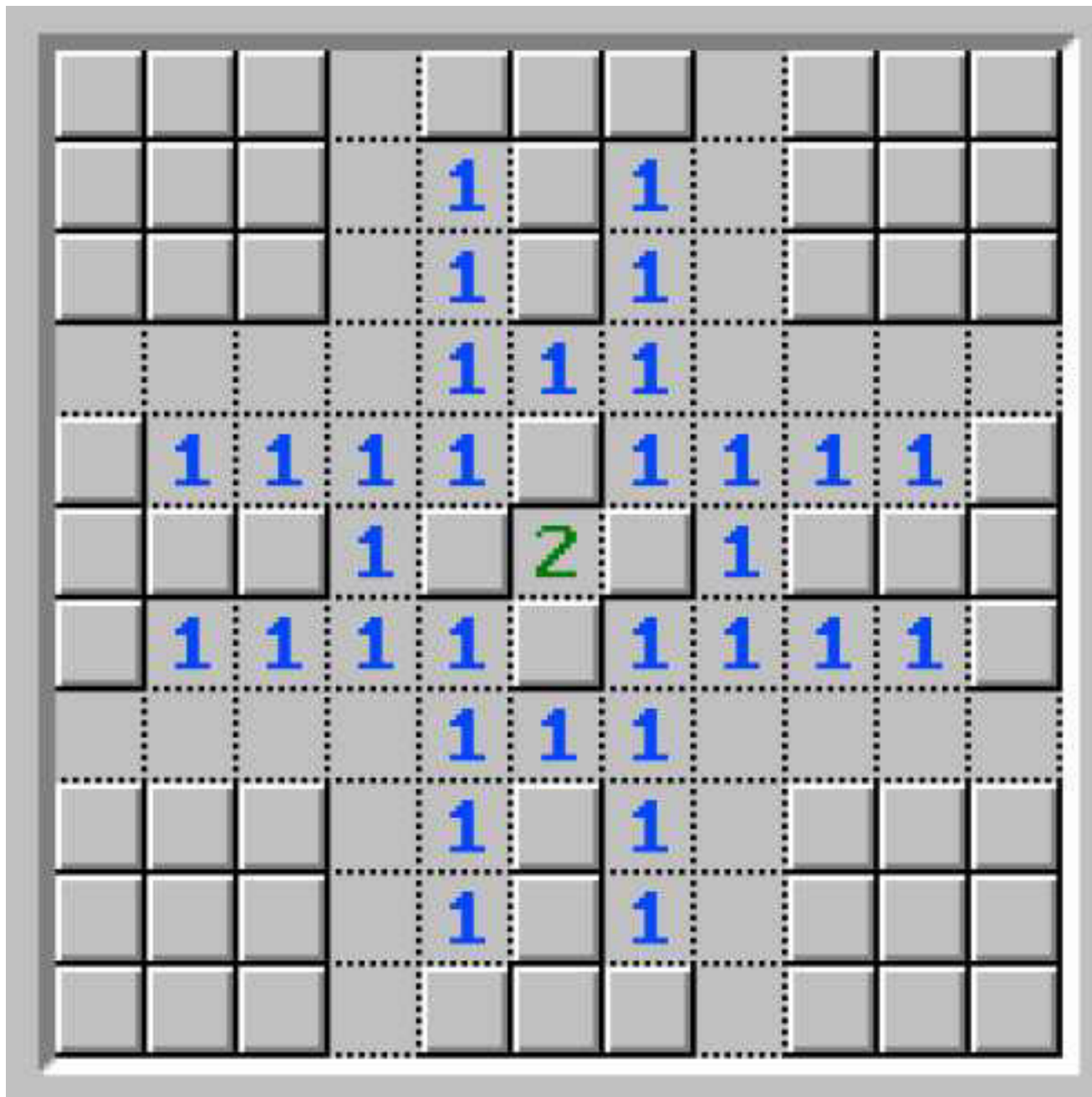
	1	1	1	1	1	1	1	1	1	1	1	1	
		1			1			1			1		
	1	1	1	1	1	1	1	1	1	1	1	1	

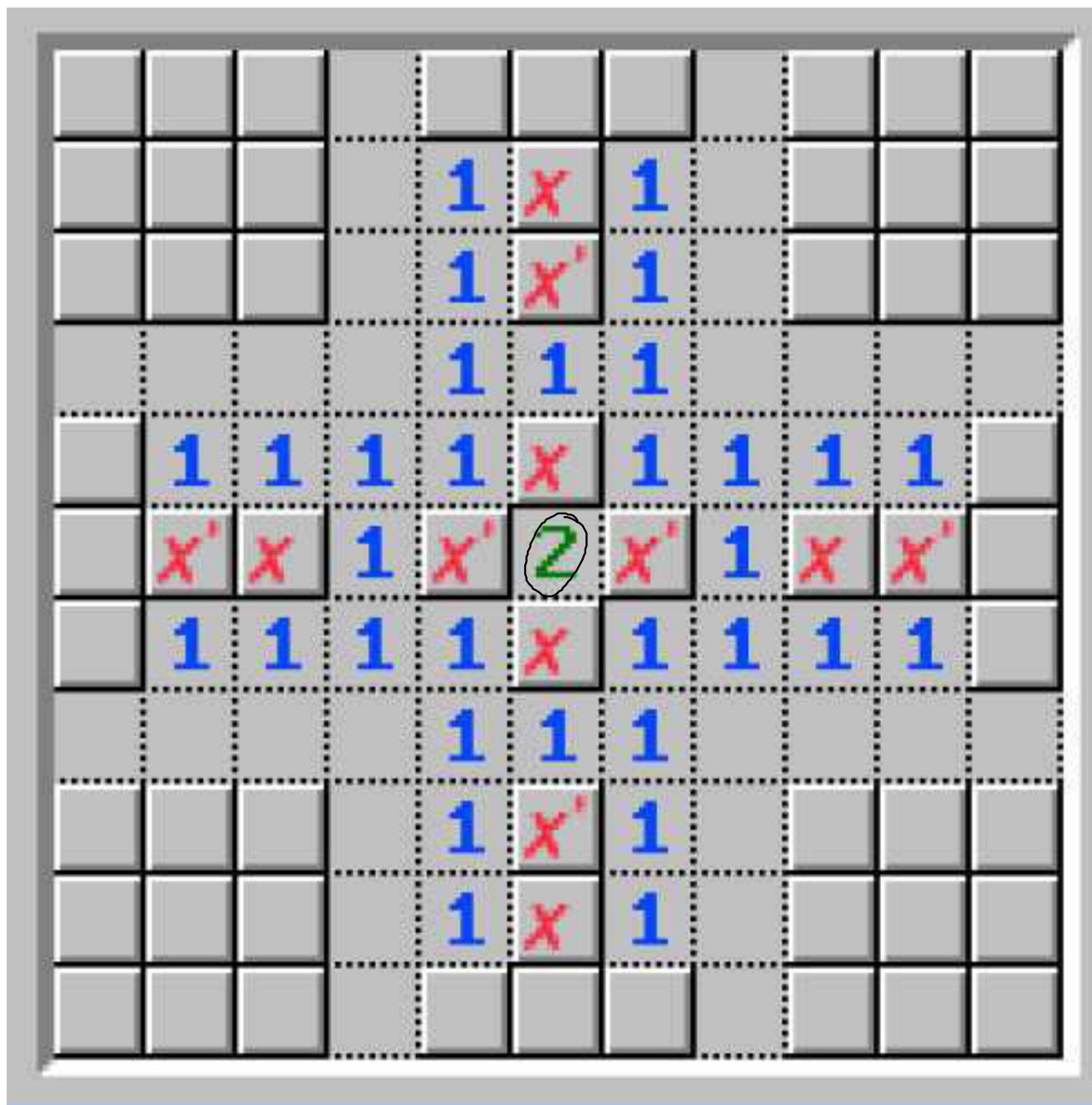




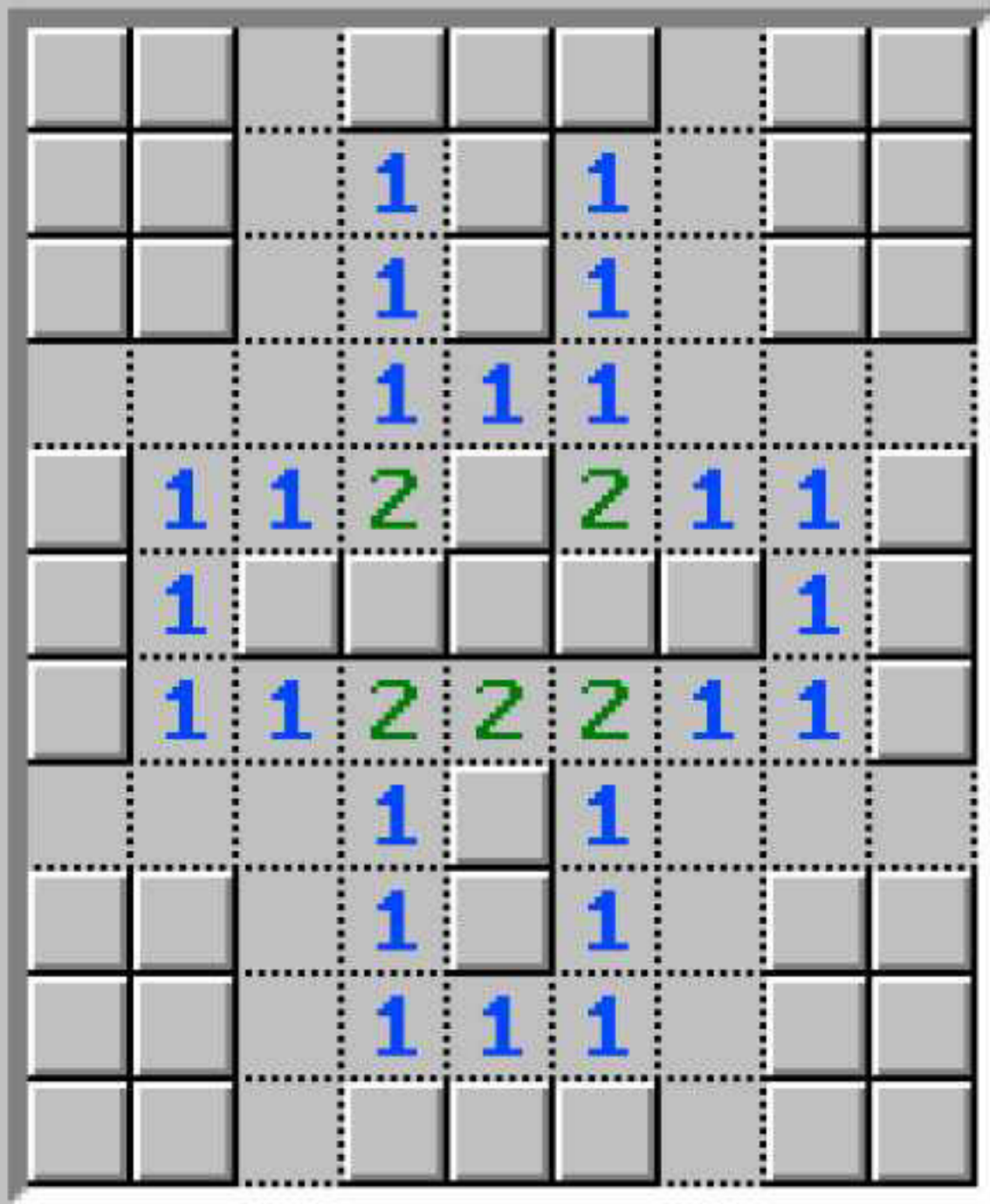
[illegible]

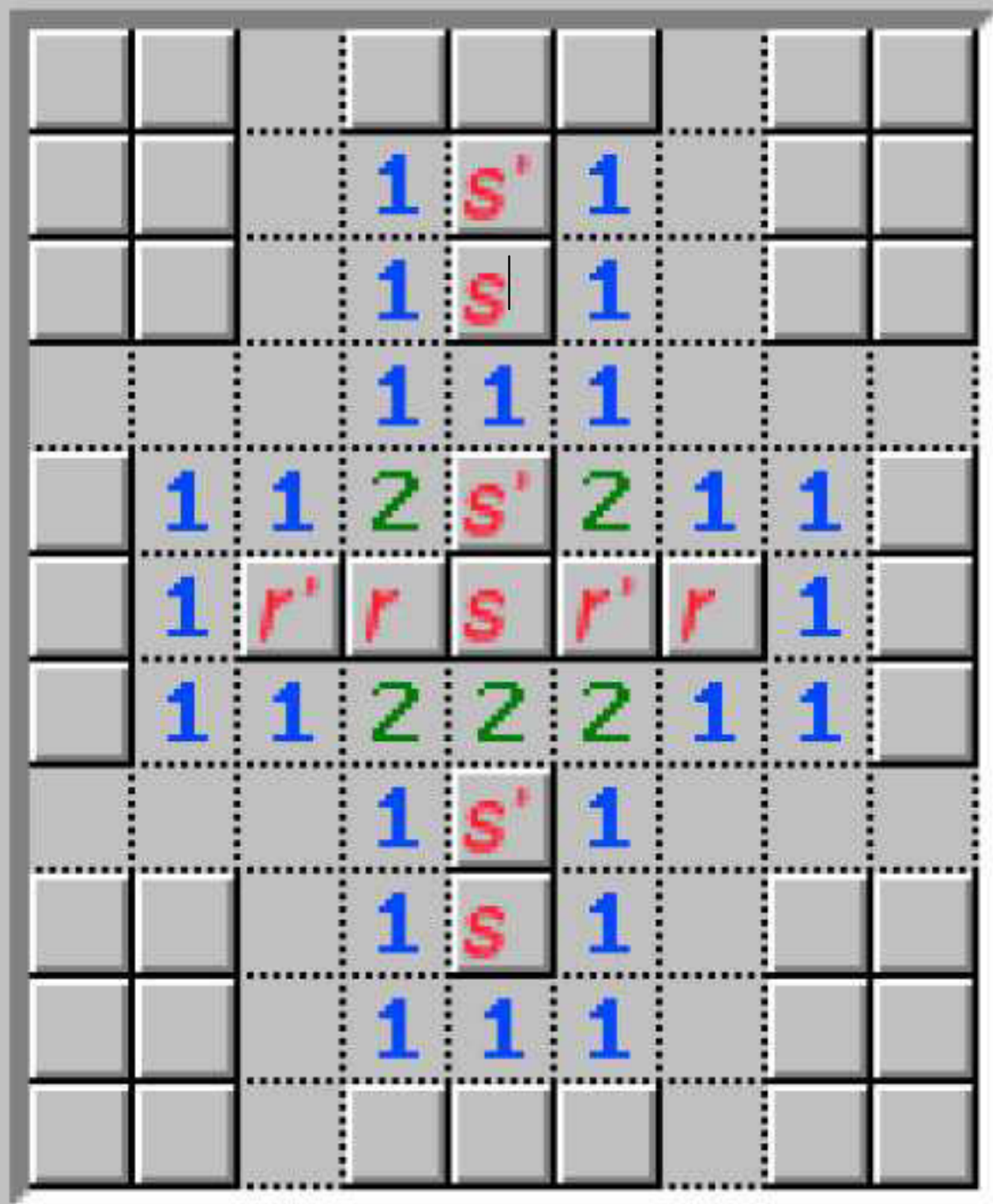
Splitter und Inverter





Übung





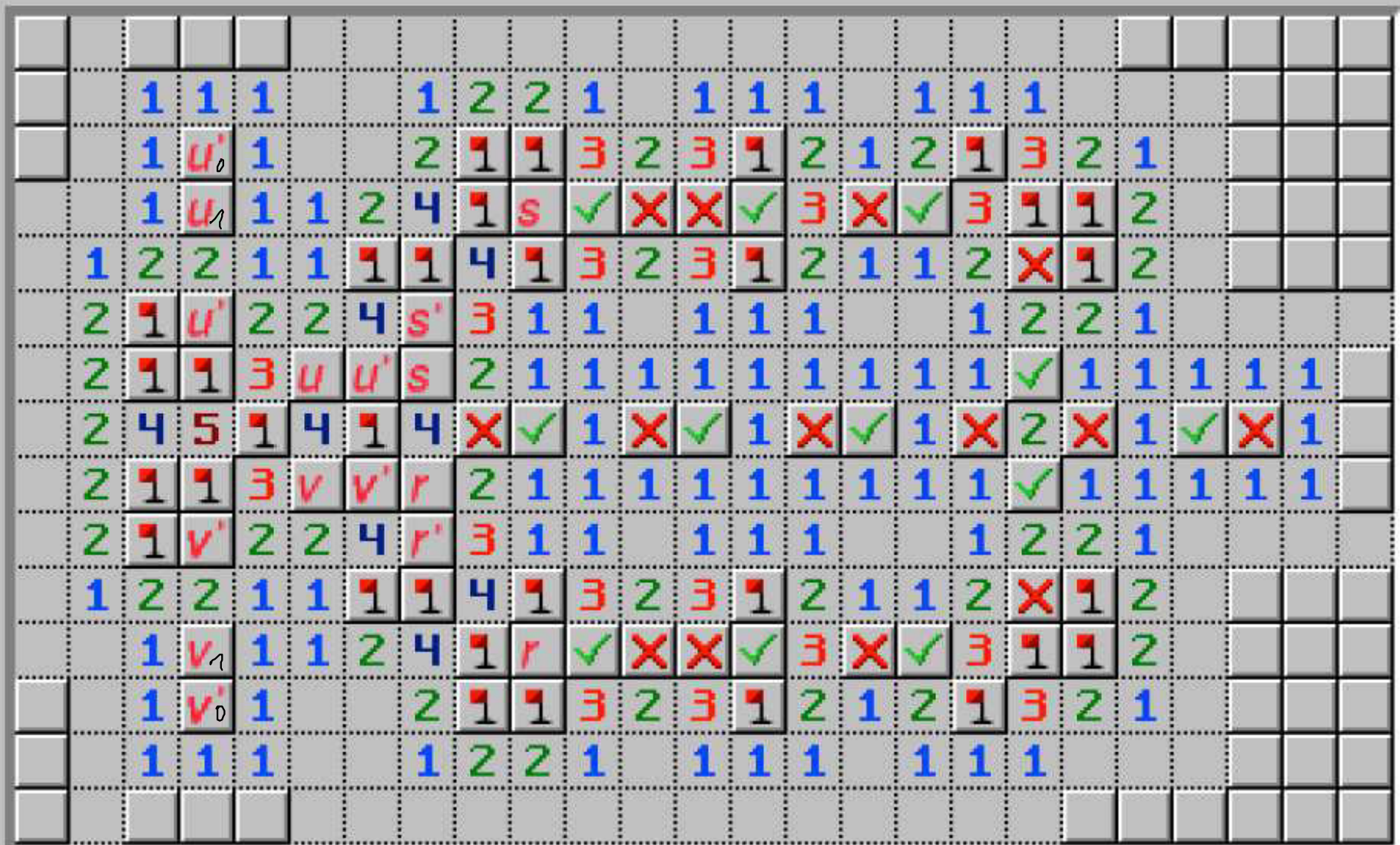
Und-Schaltkreis

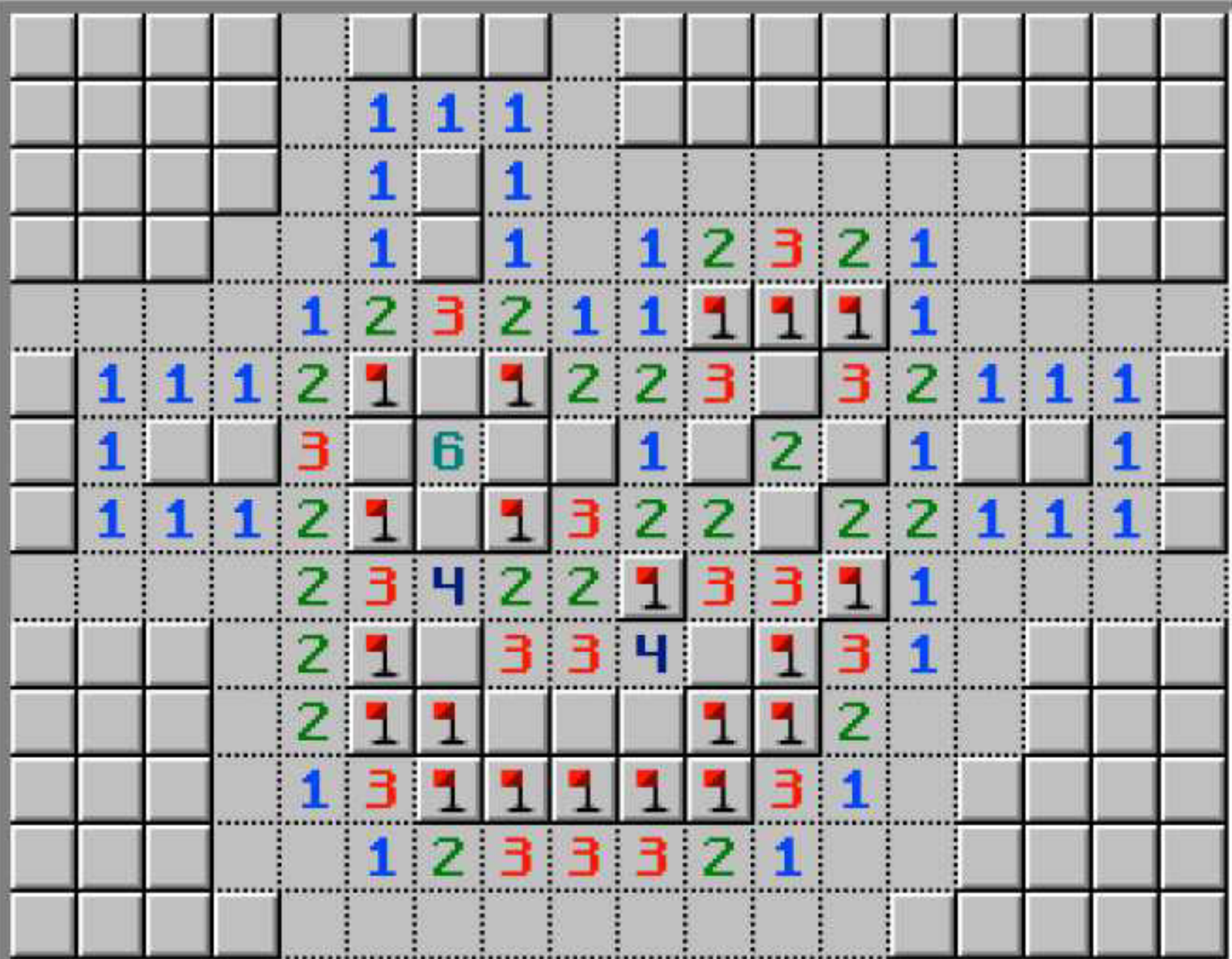


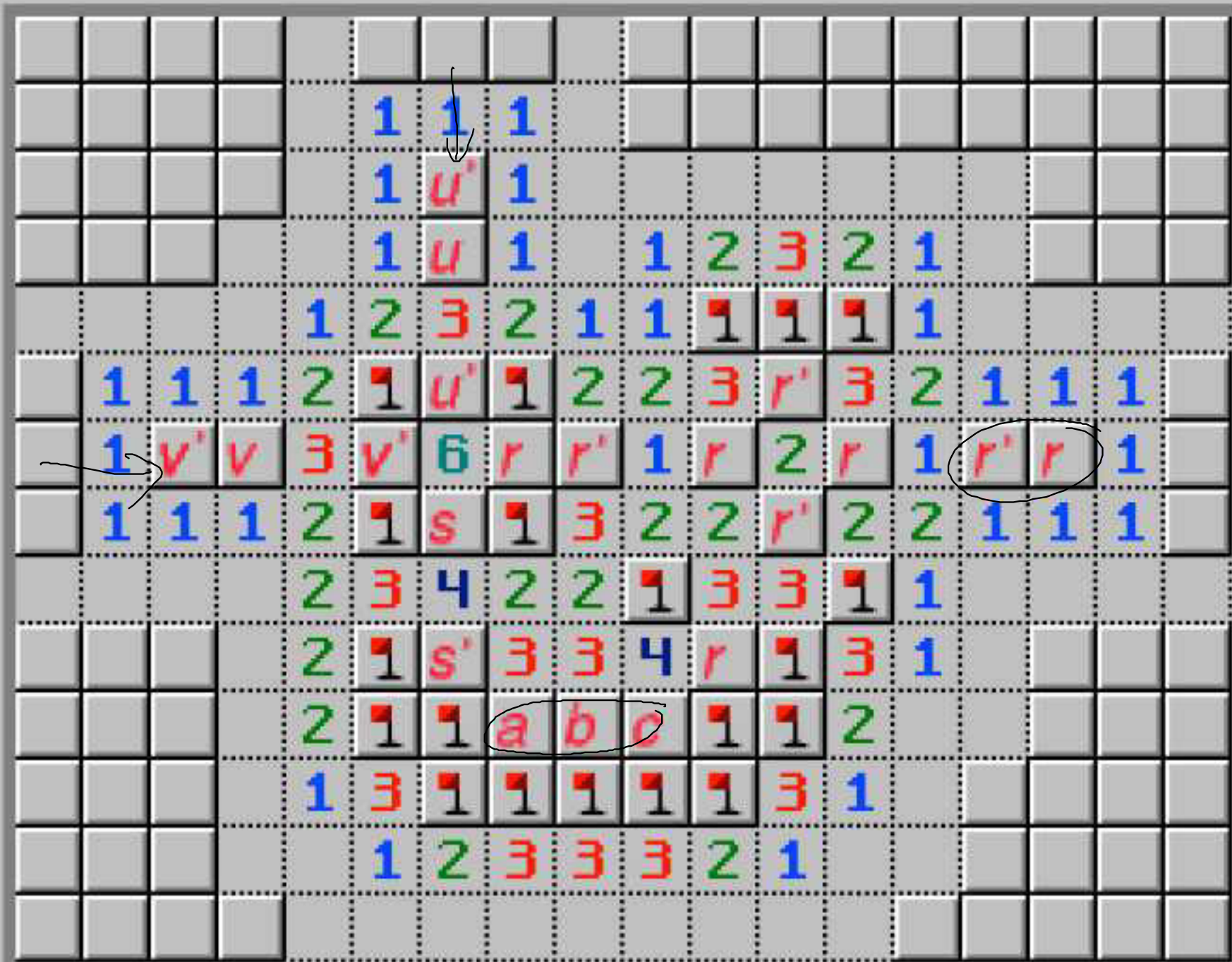
		1	1	1			1	2	2	1		1	1	1		1	1	1					
		1	u'	1			2	1	1	3	2	3	1	2	1	2	1	3	2	1			
		1	u	1	1	2	4	1	s	a	b	c	t'	3	t	t'	3	1	1	2			
	1	2	2	1	1	1	1	4	1	3	2	3	1	2	1	1	2	t	1	2			
	2	1	u'	2	2	4	s'	3	1	1		1	1	1			1	2	2	1			
	2	1	1	3	u	u'	s	2	1	1	1	1	1	1	1	1	1	t'	1	1	1	1	1
	2	4	5	1	4	1	4	t	t'	1	t	t'	1	t	t'	1	t	2	t	1	t'	t	1
	2	1	1	3	v	v'	r	2	1	1	1	1	1	1	1	1	1	t'	1	1	1	1	1
	2	1	v'	2	2	4	r'	3	1	1		1	1	1			1	2	2	1			
	1	2	2	1	1	1	1	4	1	3	2	3	1	2	1	1	2	t	1	2			
		1	v	1	1	2	4	1	r	a	b	c	t'	3	t	t'	3	1	1	2			
		1	v'	1			2	1	1	3	2	3	1	2	1	2	1	3	2	1			
		1	1	1			1	2	2	1		1	1	1		1	1	1					

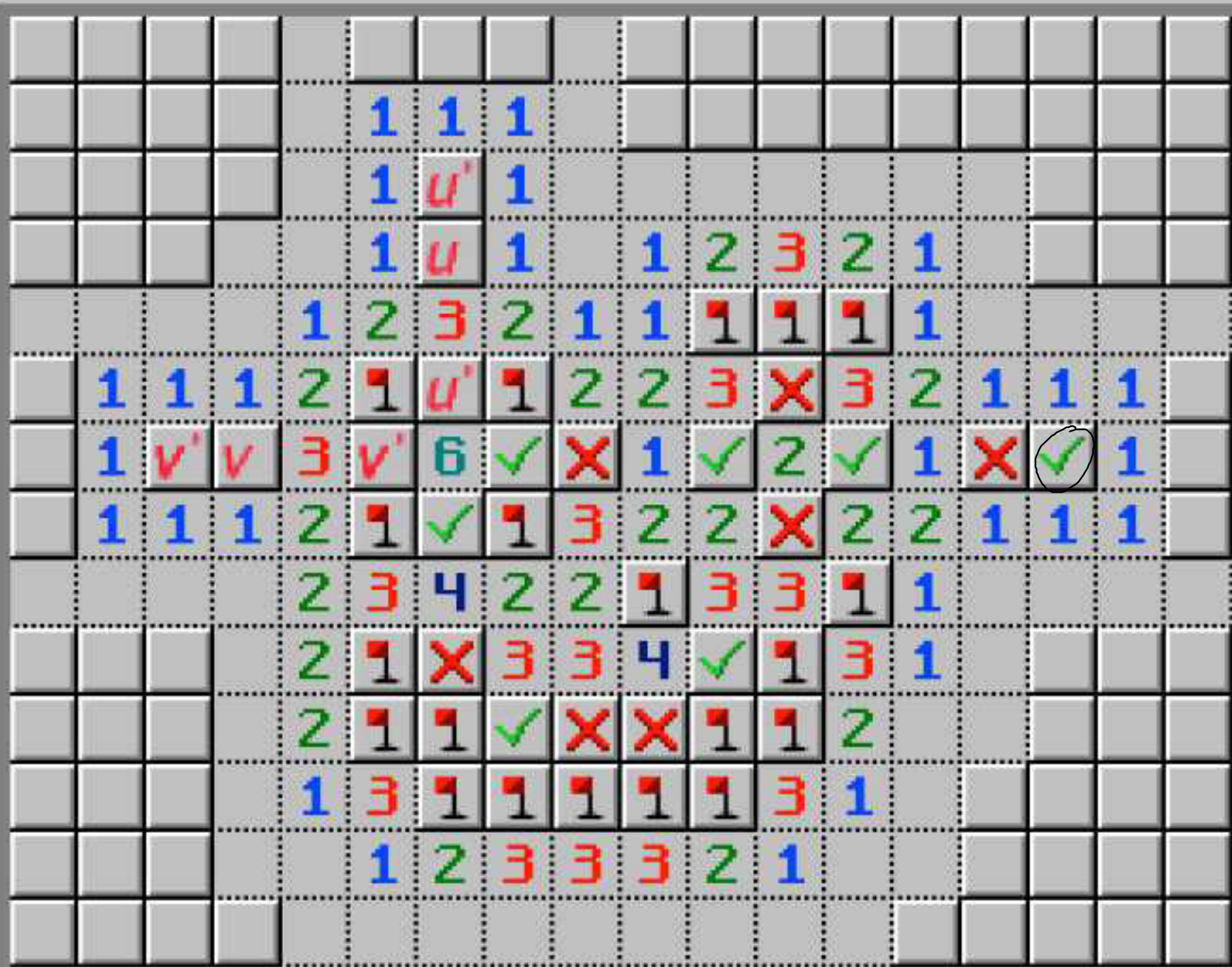
$t=0$

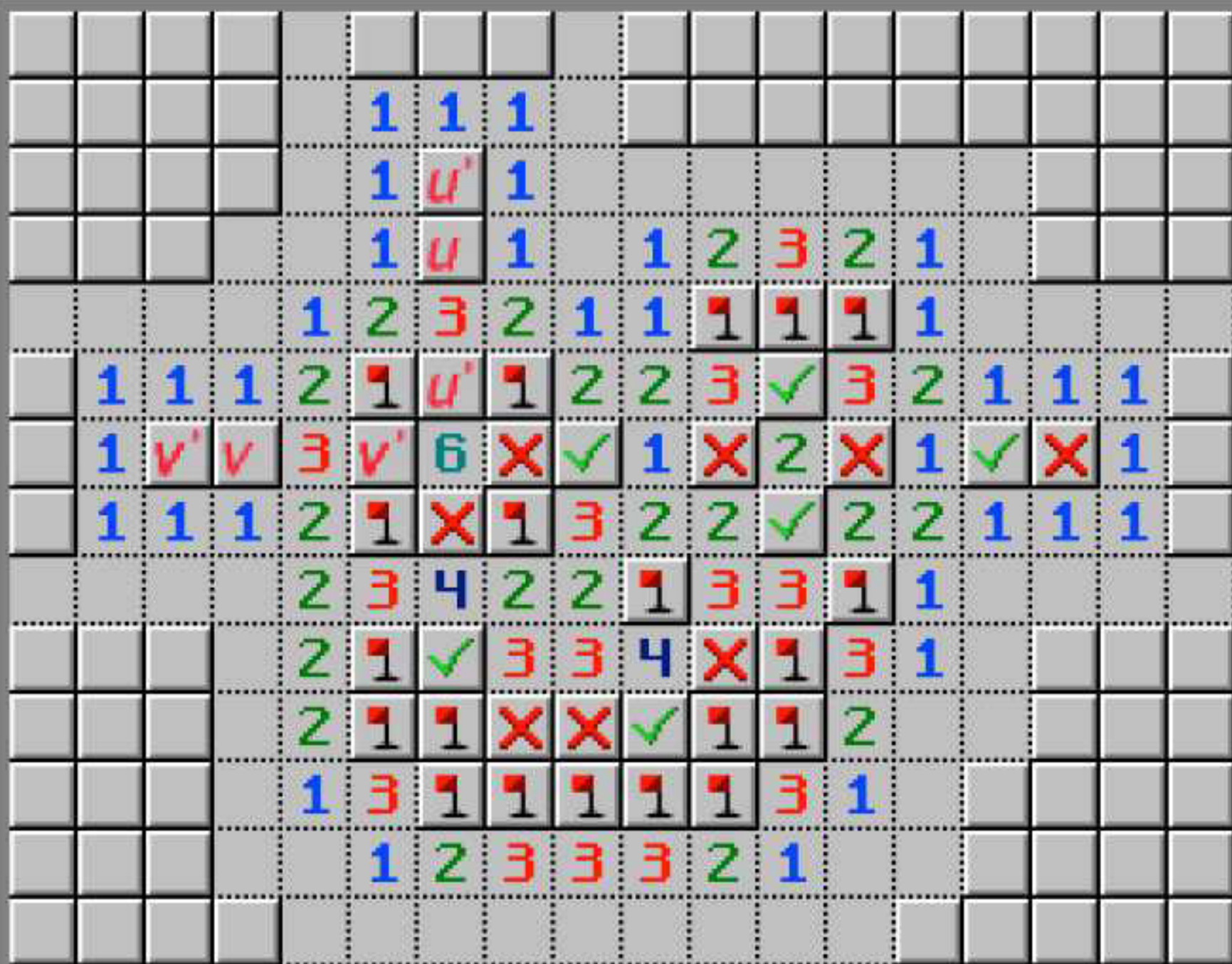

$$t = 1$$

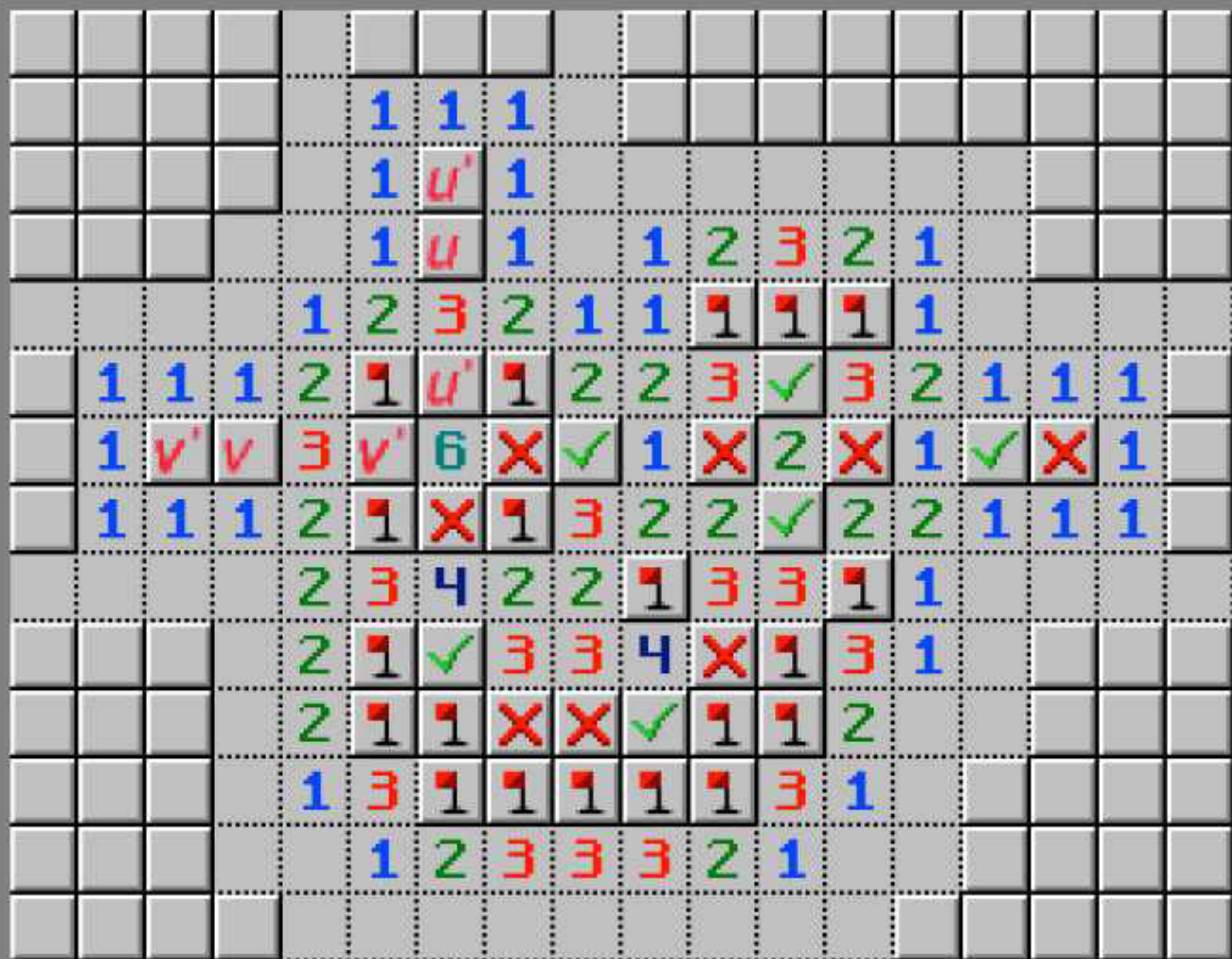












Wir können Und- und Oder-Gatter modellieren,
wir können Leitungen legen und Leitungen über Kreuzen,
wir können Leitungen vertikal und horizontal verdrahten.
D.h. wir können Schaltkreise aufbauen.

Zu jeder given Formel F können wir den
entsprechende Schaltkreis bauen, der
MSP-Instanz erzeugt und am "Ausgang" des
Schaltkreises soll eine Mine liegen ($\hat{=}$
Formel soll wahr sein), d.h.

F ist erfüllbar gdw. es ein zu der

MSP-Instanz "passendes" vollständig
aufgebautes Feld gibt.

Da die Konstruktion offenbar in poly. Zeit
funktioniert: $SAT \leq_p MSP$ $\square$

Rucksack / Knapsack - Problem

Gegeben: Eine Indexmenge $I = \{1, \dots, n\}$, eine Bewertungsfkt. $B: I \rightarrow \mathbb{N}$, eine Gewichtsfunktion $G: I \rightarrow \mathbb{N}$, ein Bewertungsminimum $b \in \mathbb{N}$ und eine Gewichtsgrenze $g \in \mathbb{N}$.

Gefragt: Ex. eine Teilmenge $J \subseteq I$, so dass

$$\sum_{j \in J} B(j) \geq b \quad \text{und} \quad \sum_{j \in J} G(j) \leq g \quad ?$$

SUBSET-SUM

Gegeben: $a_1, \dots, a_n \in \mathbb{N}$ und $b \in \mathbb{N}$.

Gefragt: Gibt es $J \subseteq \{1, \dots, n\}$ mit $\sum_{j \in J} a_j = b$?

Bem: Subset-Sum ergibt sich aus Knapsack für $b = g$ und $B = G$.

Satz SUBSET-SUM ist NP-vollständig.

$\in NP$: Ja, kann $\exists$ ed verifiziert.

NP-Härte: Reduktion von 3CNF-SAT auf SUBSET-SUM.

$$S_n: F = (l_{n1} \vee l_{n2} \vee l_{n3}) \wedge \dots \wedge (l_{m1} \vee l_{m2} \vee l_{m3})$$

$$\text{mit } l_{ij} \in \{x_1, \dots, x_n\} \cup \{\neg x_1, \dots, \neg x_n\}$$

(l_{ij} Klausel und x_i Var.)

Konstruiere SUBSET-SUM-Instanz:

$$b = \underbrace{44 \dots 4}_m \underbrace{11 \dots 1}_n$$

= Anzahl der Klauseln Anzahl der Var.

$$(x_1 \vee \neg x_1) \wedge (\neg x_1 \vee x_2 \vee x_3)$$

$$b = 44 \ 111$$

$$v_1 = 10 \ 100 \quad v_1' = 01 \ 100$$

$$v_2 = 01 \ 010 \quad v_2' = 10 \ 010$$

$$v_3 = 01 \ 001 \quad v_3' = 00 \ 001$$

$$\begin{array}{r} 01 \ 100 \\ 10 \ 010 \\ 01 \ 001 \\ \hline 1 \ 2 \ 111 \\ + c_1 + d_1 + d_2 \end{array}$$

$a_1, \dots, a_n$: Dafür konstruiere wir 4 verschiedene Gruppen
 $v_i, v_i', c_j, d_j \quad i \in \{1, \dots, n\}, j \in \{1, \dots, m\}$

Für positives Vorkommen von x_i in Klauseln j_1, j_2, j_3
m Ziffern n Ziffern

$$v_i = 0 \dots 010 \dots 010 \dots 010 \dots 000 \dots 010 \dots 000 \leftarrow$$

j_1 te Stelle j_2 te Stelle j_3 te Stelle i te Stelle

Für negatives Vorkommen von x_i

$$v_i' = 0 \dots 010 \dots 010 \dots 010 \dots 000 \dots 010 \dots 000 \leftarrow$$

$$c_j = \underbrace{00 \dots 1 \dots 00 \dots 0}_m \underbrace{0 \dots 0 \dots 0}_n$$

j te Stelle

$$d_j = 00 \dots 0 \dots 00 \dots 00 \dots 0$$

Nehme wir an, es gibt J mit $\sum_{j \in J} a_j = b$.

Dann muss für jedes x_i entweder v_i oder v_i' gewählt werden sein ($\pm$ Entsprechung pos./neg. Vorkommen in Klausel k). Wir müssen bei der ersten Stelle immer mindestens eine "1" haben ($\pm$ erfüllte Klausel). D.h. aus der Menge J können wir eine erfüllende Belegung für F erzeugen.

Nehme wir an F ist erfüllbar. Dann erhält man eine Indexmenge J' entsprechend der erfüllenden Belegung, so dass $b = x_1 x_2 \dots x_m \vee_1 \dots \vee_n$ mit $\vee_i = 1$ und $x_j \geq 1$. Jetzt kann man J' und c_j und d_j erweitern um b zu erweitern, so dass das

$$\sum_{j \in J} a_j = b \text{ gilt.}$$

□

Dynamic Programming - Alg.