Reine Mathematik: Alles nur Spielerei!

Prof. Dr. Ulrich Thiel

https://ulthiel.com/math

RPTU Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

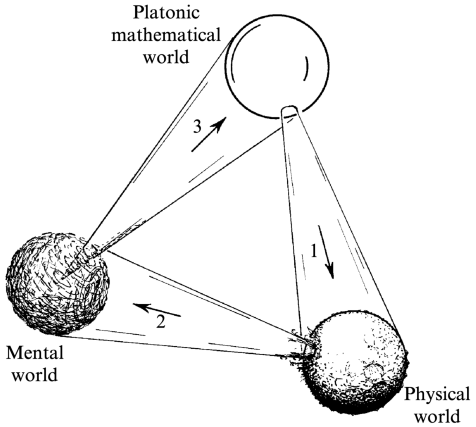
1. Einleitung: Reine Mathematik

Ich: Professor für Algebra, d.h. für reine Mathematik ("abstrakter Unsinn")

(Nennt man scherzhaft / wirklich so!)

"Rein" heißt, dass man mathematische Zusammenhänge um ihrer selbst Willen studiert und nicht unbedingt Anwendungen außerhalb der Mathematik (z.B. die "reale Welt") im Fokus hat.

Bemerkung: Dennoch sind viele Strukturen in der reinen Mathematik durch reale Dinge motiviert und manchmal finden sie später den Weg zurück zur Realität!
z.B. Symmetrie ⇝ Vermutung der Existenz des Higgs-Bosons ⇝ CERN findet es!



(Penrose: The road to reality)

Fragen: Warum ist es "erlaubt", sich mit reinen Spieleseien zu beschäftigen?
Haben wir nicht wichtigere Probleme (z.B. Klimawandel)?
Warum sollten Sie sich damit beschäftigen?

Meine Antwort: Was wir brauchen sind Leute, die:

* analytisch, ausdauernd, selbständig, kritisch denken können
* sachliche Diskussionen führen können, mit Kritik umgehen können
* hohe Frustrationstoleranz haben

} allgemeine Fähigkeiten statt spezielles Fachwissen!

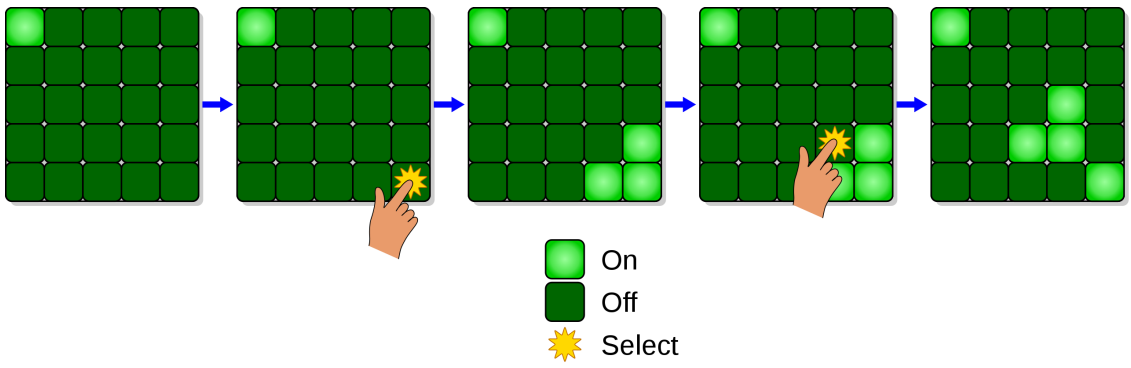
Genau diese Fähigkeiten trainiert man im Mathematikstudium, besonders (natürlich...) in der reinen Mathematik!

2. Ein Spiel

Das Spiel habe ich mir nicht selbst ausgedacht, sondern ↙ Tiger Electronics, 1995 (in dieser Form)

Da Spielerei nun erlaubt ist, spielen wir ein Spiel!
Es führt uns zu Strukturen, die man im 1. Semester behandelt.

Stellen Sie sich ein 5x5-Gitter aus Leuchtknöpfen vor.
Eine zufällige Ansammlung davon leuchtet.
Drückt man einen Knopf, so wir das Licht dieses Knopfes und seiner direkten horizontalen und vertikalen Nachbarn umgeschaltet, also an → aus und aus → an.



Quelle: Wikipedia

Ziel des Spiels: Alle Lichter ausschalten!

Selbst ausprobieren:
https://www.jaapsch.net/puzzles/lights.htm

Frage: Ist das für eine gegebene Startkonfiguration überhaupt möglich?
(Als Mathematiker/Mathematikerin fragt man sich sowas sofort! → Training!)

Vielleicht fallen Ihnen noch weitere Fragen ein?

Quizfrage: Wieviele Startkonfigurationen gibt es eigentlich?

3. Formales Modellieren: Konfiguration

Den Zustand eines Knopfes können wir mit 0="Licht aus" und 1="Licht an" kodieren.

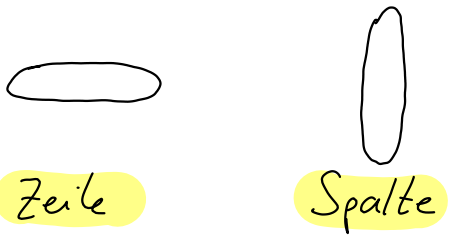
Die Konfiguration des gesamten Gitters können wir mit einem Vektor kodieren:

einfach eine Zahlenkolonne

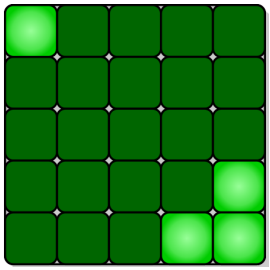
$$\vec{b} = (b_{(1,1)}\ b_{(1,2)}\ b_{(1,3)}\ b_{(1,4)}\ b_{(1,5)}\ b_{(2,1)}\ b_{(2,2)} \cdots b_{(5,5)})^T \in \{0,1\}^{25}$$

"transponiert", d.h. als Zeile geschrieben

$b_{(i,j)}$ = Zustand in Zeile i und Spalte j



Beispiel:



$\vec{b} = (10000\ 00000\ 00000\ 00001\ 00011)^T$

Beispiel: Alle Lichter aus (das Ziel) entspricht dem Nullvektor

$$\vec{0} = (0\,0\ldots0)^T$$

4. Formales Modellieren: Operation

Wie modellieren wir die Operation $\vec{b} \rightsquigarrow \vec{c}$ des Drückens eines Knopfes?

Wir machen dazu etwas "Verrücktes". Wir definieren + und · auf $\{0,1\}$ neu:

+	0	1
0	0	1
1	1	0 ← d.h. 1+1=0!

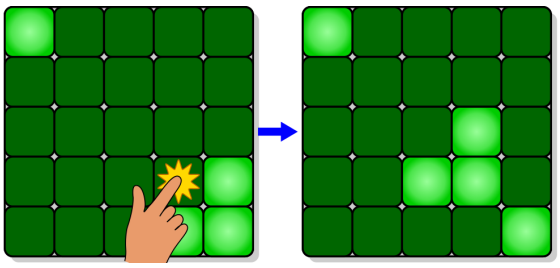
·	0	1
0	0	0
1	0	1

Dieses + und · erfüllt alle "üblichen" Rechenregeln: Kommutativgesetz, Assoziativgesetz, Distributivgesetz, man kann Subtrahieren und Dividieren, ...

In der Mathematik sagt man: $\mathbb{F}_2 := (\{0,1\}, +, \cdot)$ ist ein Körper.
Ein Körper ist einfach eine (alternative) Welt zum Rechnen!
z.B. $\mathbb{Q}, \mathbb{R}, \mathbb{C}$ sind Körper; aber auch $\mathbb{F}_2$. (Warum ist $\mathbb{Z}$ kein Körper?)

Das Drücken eines Knopfes ändert die Konfiguration $\vec{b}$ durch (komponentenweise) Addition eines Vektors, der den Eintrag 1 an der Position des Knopfes und seiner direkten Nachbarn hat und der sonst überall den Eintrag 0 hat.

Beispiel:



$$\begin{pmatrix}1&0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0&1\end{pmatrix}\begin{pmatrix}0&0&0&1&1\end{pmatrix}^T$$ Ausgangskonfiguration

$$+\begin{pmatrix}0&0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&1&0\end{pmatrix}\begin{pmatrix}0&0&1&1&1\end{pmatrix}\begin{pmatrix}0&0&0&1&0\end{pmatrix}^T$$ Knopf (4,4) drücken

$$=\begin{pmatrix}1&0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&0&0\end{pmatrix}\begin{pmatrix}0&0&0&1&0\end{pmatrix}\begin{pmatrix}0&0&1&1&0\end{pmatrix}\begin{pmatrix}0&0&0&0&1\end{pmatrix}^T$$ Neue Konfiguration

5. Formales Modellieren: Strategien und Lösbarkeit

Zwei wichtige Beobachtungen:

1. Drückt man mehrere Knöpfe hintereinander, so ist das Ergebnis unabhängig von der Reihenfolge, in des man die Knöpfe drückt.
   Das liegt einfach daran, dass + das Kommutativgesetz und das Assoziativgesetz erfüllt!

2. Drückt man einen Knopf zweimal, so ist das als ob man den Knopf gar nicht gedrückt hätte.
   Das liegt an $1+1=0$!
   D.h. es kommt nur darauf an, ob ein Knopf gedrückt wird oder nicht.

Folgerung: Wir können eine Abfolge von Knöpfe-Drücken (Strategie) durch einen Vektor

$$\vec{x} = (x_{(1,1)}\ x_{(1,2)}\ x_{(1,3)}\ x_{(1,4)}\ x_{(1,5)}\ x_{(2,1)}\ x_{(2,2)}\ \cdots\ x_{(5,5)})^T \in \{0,1\}^{25}$$

kodieren, wobei $x_{i,j} = 1$ falls der Knopf (i,j) gedrückt wurde, alle anderen Einträge $= 0$.

Wir nennen eine Konfiguration $\vec{b}$ lösbar, wenn eine Strategie $\vec{x}$ existiert, die auf $\vec{b}$ angewandt alle Lichter ausschaltet, d.h. den Nullvektor $\vec{0}$ produziert.

Beobachtung: Entsteht $\vec{b}$ aus $\vec{0}$ durch $\vec{x}$, so erzeugt nochmaliges Anwenden von $\vec{x}$ wieder $\vec{0}$ (wegen $1+1=0$!)
D.h. die lösbaren Konfigurationen sind genau die, die man durch beliebige Strategien aus $\vec{0}$ erhält!

6. Ergebnis einer Strategie

Sei $\vec{b}$ die Konfiguration, die man aus der Strategie $\vec{x}$ angewandt auf $\vec{0}$ erhält. Dann gilt:

$$\begin{aligned}
b_{(1,1)} &= x_{(1,1)} + x_{(1,2)} + x_{(2,1)} \\
b_{(1,2)} &= x_{(1,1)} + x_{(1,2)} + x_{(1,3)} + x_{(2,2)} \\
b_{(1,3)} &= x_{(1,2)} + x_{(1,3)} + x_{(1,4)} + x_{(2,3)}
\end{aligned}$$

usw.

In der Gleichung für $b_{(i,j)}$ stehen einfach als Summe alle $x_{k,\ell}$, sodass das Drücken des Knopfes (k,ℓ) sich auf (i,j) auswirkt, d.h. die (k,ℓ) sind die direkten Nachbarn von (i,j), inklusive (i,j).

Man kann das kompakt schreiben als

$$\vec{b} = \begin{pmatrix}
1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&1&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&1&1&1&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&1&1&1&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1
\end{pmatrix} \cdot \vec{x}$$

← Man nennt sowas eine Matrix

· $\vec{x}$: d.h. erste Spalte · $x_{(1,1)}$ + zweite Spalte · $x_{(1,2)}$ usw.

Wir bezeichnen die Matrix mit A. Es gilt also $\vec{b} = A \cdot \vec{x}$.

7. Lösbarkeit

Unser Ergebnis: $\vec{b}$ lösbar genau dann, wenn $\vec{b} = A\vec{x}$ für ein $\vec{x} \in \{0,1\}^{25}$.

(wichtige Formulierung in der Mathematik)

Das ist einfach ein lineares Gleichungssystem in $\vec{x}$!

Das kann man in 25 Dimensionen und über $\mathbb{F}_2$ ($1+1=0$) mit der gleichen Methode lösen wie man das in der Schule in 2 oder 3 Dimensionen über $\mathbb{R}$ gemacht hat!

↝ Gaußsches Eliminationsverfahren; funktioniert über jedem Körper

↝ 1. Semester

Wir können so nicht nur testen, ob $\vec{b}$ lösbar ist, sondern finden auch gleich eine Strategie $\vec{x}$, falls $\vec{b}$ lösbar ist!

Nur in der Schule und im Studium löst man lineare Gleichungssysteme per Hand. Wir nehmen den Computer!

8. Computerhilfe

Die **Algebra-Gruppe** an der RPTU ist international herausragend auf dem Gebiet des **Computeralgebra**.

Z.B entwickeln wir **OSCAR**:

Open Source Computer Algebra Research

OSCAR SYMBOLIC TOOLS
https://www.oscar-system.org

Ein neues Computeralgebra-System basierend auf der modernen high performance Programmiersprache **Julia**.
https://www.julialang.org
(Einfacher Syntax wie Python aber Performance wie C)

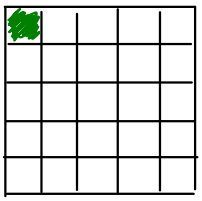
Wir starten Julia und laden OSCAR:

```
~ $julia

               _
   _       _ _(_)_     |  Documentation: https://docs.julialang.org
  (_)     | (_) (_)    |
   _ _   _| |_  __ _   |  Type "?" for help, "]?" for Pkg help.
  | | | | | | |/ _` |  |
  | | |_| | | | (_| |  |  Version 1.8.5 (2023-01-08)
 _/ |\__'_|_|_|\__'_|  |  Official https://julialang.org/ release
|__/                   |

julia> using Oscar
 -----    -----    -----      -      -----
|     |  |     |  |     |    | |    |     |
|     |  |        |         |   |   |     |
|     |   -----   |        |     |  |-----
|     |        |  |        |-----|  |   |
|     |  |     |  |     |  |     |  |    |
 -----    -----    -----   -     -  -     -

...combining (and extending) ANTIC, GAP, Polymake and Singular
Version 0.12.2-DEV ...
 ... which comes with absolutely no warranty whatsoever
Type: '?Oscar' for more information
(c) 2019-2023 by The OSCAR Development Team

julia>
```

Nun bauen wir die Matrix A über dem Körper $\mathbb{F}_2$ (mit ein paar Tricks, um Schreibarbeit zu sparen):

```
julia> K = GF(2);

julia> B = matrix(K, 5, 5, [1,1,0,0,0, 1,1,1,0,0, 0,1,1,1,0, 0,0,1,1,1, 0,0,0,1,1]);

julia> I = identity_matrix(K, 5);

julia> O = zero_matrix(K, 5, 5);

julia> A = vcat(map(hcat, [B,I,O,O,O], [I,B,I,O,O], [O,I,B,I,O], [O,O,I,B,I], [O,O,O,I,B]));

julia> A
[1   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[1   1   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[0   1   1   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[0   0   1   1   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[0   0   0   1   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[1   0   0   0   0   1   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0]
[0   1   0   0   0   1   1   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0]
[0   0   1   0   0   0   1   1   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0]
[0   0   0   1   0   0   0   1   1   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0]
[0   0   0   0   1   0   0   0   1   1   0   0   0   0   1   0   0   0   0   0   0   0   0   0]
[0   0   0   0   0   1   0   0   0   1   1   0   0   0   1   0   0   0   0   0   0   0   0   0]
[0   0   0   0   0   0   1   0   0   0   1   1   1   0   0   0   1   0   0   0   0   0   0   0]
[0   0   0   0   0   0   0   1   0   0   0   1   1   1   0   0   0   1   0   0   0   0   0   0]
[0   0   0   0   0   0   0   0   1   0   0   0   1   1   0   0   0   0   1   0   0   0   0   0]
[0   0   0   0   0   0   0   0   0   1   0   0   0   0   1   1   0   0   0   1   0   0   0   0]
[0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   1   1   0   0   0   1   0   0   0]
[0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   1   1   0   0   0   1   0   0]
[0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   1   1   0   0   0   1   0]
[0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   1   0   0   0   0   1]
[0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   1   1   0   0   0]
[0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   1   1   0   0]
[0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   1   1   0]
[0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   1   1]
[0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   1]
```

Frage: Ist die Konfiguration



Lösbar? Wir definieren dazu den entsprechenden Vektor $\vec{b}$:

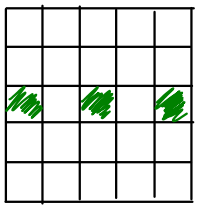
```
julia> b = matrix(K, 1, 25, [1,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0, 0,0,0,0,0])
[1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
```

Und nun testen wir, ob $A\vec{x} = \vec{b}$ eine Lösung hat:

```
julia> can_solve_with_solution(A, transpose(b), side = :right)
(false, [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0])
```

Die Antwort ist nein!

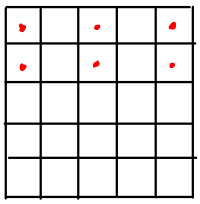
Frage: Ist die Konfiguration



Lösbar?

```
julia> b = matrix(K, 1, 25, [0,0,0,0,0, 0,0,0,0,0, 1,0,1,0,1, 0,0,0,0,0, 0,0,0,0,0])
[0   0   0   0   0   0   0   0   0   0   1   0   1   0   1   0   0   0   0   0   0   0   0   0   0]

julia> can_solve_with_solution(A, transpose(b), side = :right)
(true, [1; 0; 1; 0; 1; 1; 0; 1; 0; 1; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0])
```

Die Antwort ist ja! Und wir bekommen auch eine Lösung $\vec{x}$ und damit eine Strategie angezeigt. Folgende Knöpfe muss man drücken:



Wir können auch ein paar allgemeine Fragen zum Spiel beantworten.

Frage: Wenn eine Lösung existiert, ist diese eindeutig?

Ist $\vec{b} = A\vec{x}$ und $\vec{y}$ mit $A\vec{y} = \vec{0}$, so ist

$$A(\vec{x}+\vec{y}) = A\vec{x} + A\vec{y} = \vec{b} + \vec{0} = \vec{b},$$

d.h. $\vec{x}+\vec{y}$ ist auch eine Lösung. Ist $\vec{b} = A\vec{x} = A\vec{x'}$, so ist

$$\vec{0} = \vec{b} - \vec{b} = A\vec{x} - A\vec{x'} = A(\vec{x}-\vec{x'}),$$

d.h.

$$\vec{x}-\vec{x'} \in \mathrm{Kern}(A) := \{\vec{y} \in \{0,1\}^{25} \mid A\vec{y} = \vec{0}\}$$

$:=$ heißt "wird definiert als".

Also, Kern(A) bestimmt die Anzahl der Lösungen.

Wir berechnen Kern(A) wieder im Computer:

```
julia> transpose(nullspace(A)[2])
[0  1  1  1  0  1  0  1  0  1  1  1  0  1  1  1  0  1  0  1  0  1  1  1  0]
[1  0  1  0  1  1  0  1  0  1  0  0  0  0  0  1  0  1  0  1  1  0  1  0  1]
```

Der Computer gibt eine **Basis** von Kern(A) aus, d.h. Kern(A) besteht aus allen **Linearkombinationen** (↑ alle möglichen Summen) der Basis-Elemente.

Hier besteht die Basis aus 2 Elementen $\vec{y}_1$ und $\vec{y}_2$.

Wegen $\mathbb{F}_2 = \{0,1\}$ besteht Kern(A) also aus:

$$\vec{0},\ \vec{y}_1,\ \vec{y}_2,\ \vec{y}_1 + \vec{y}_2$$

Folgerung: Ist $\vec{b}$ lösbar, so gibt es genau **4** Strategien!

Frage: Wieviele der $2^{25} = 33.554.432$ Konfigurationen besitzen eine Lösung?

Wir haben gezeigt: $\vec{b} \in \{0,1\}^{25}$ ist lösbar, genau dann, wenn

$$\vec{b} \in \text{Bild}(A) := \{ A\vec{x} \mid \vec{x} \in \{0,1\}^{25} \}.$$

Mit dem Gauß-Verfahren kann man auch eine Basis von Bild(A) berechnen. Die Anzahl der Basis-Vektoren nennt man den **Rang** von A.

Wir fragen den Computer:

```
julia> rank(A)
23
```

(mit der Theorie im 1. Semester folgt das bereits aus der Dimension für den Kern: $25 = 2 + 23$)

Da Bild(A) aus den Linearkombinationen der Basisvektoren besteht und $\mathbb{F}_2 = \{0,1\}$, folgt

(Anzahl der Elemente →) $\#\text{Bild}(A) = 2^{23}$

Es sind also $\frac{2^{23}}{2^{25}} = 2^{-2} = \frac{1}{4} \hat{=} 25\%$ aller Konfigurationen lösbar!

9. Ein effizienter Test

Um zu testen, ob eine Konfiguration lösbar ist, muss man (bisher) das lineare Gleichungssystem $A\vec{x} = \vec{b}$ lösen. Es gibt eine effizientere Methode.

Nach unserer Diskussion sind die lösbaren Konfigurationen genau die Elemente in

$$\text{Bild}(A) = \left\{ A\vec{x} \mid \vec{x} \in \{0,1\}^{25} \right\}.$$

Nach Definition von $A \cdot \vec{x}$ gilt

$$\text{Bild}(A) = \left\{ \text{erste Spalte} \cdot x_{(1,1)} + \text{zweite Spalte} \cdot x_{(1,2)} + \dots \mid \vec{x} \in \{0,1\}^{25} \right\}$$

d.h. Bild(A) besteht aus allen möglichen Linearkombinationen der Spalten.

Sei $\vec{y} = A\vec{x}$. Dann gilt nach Definition von $A \cdot \vec{x}$ für die Einträge von $\vec{y}$:

$$y_{(i,j)} = \sum_{k,\ell=1}^{5} b_{(k,\ell)} \cdot x_{(k,\ell)}$$

, wobei $\vec{b}$ die Zeile mit "Nummer" (i,j) von A ist

$$=: \vec{b} \cdot \vec{x}$$

↑ eine Art "Skalarprodukt"

Dann gilt für $\vec{x} \in \text{Kern}(A)$, d.h. $A\vec{x} = 0$, also

$$\vec{b} \cdot \vec{x} = 0 \quad \text{für alle Zeilen } \vec{b} \text{ von } A.$$

Beobachtung: A ist symmetrisch, d.h. Spalte i = Zeile i für alle i.

Daher gilt

$$\text{Bild}(A) = \left\{ (\text{erste Zeile})^T \cdot x_{(1,1)} + (\text{zweite Zeile})^T \cdot x_{(1,2)} + \dots \mid \vec{x} \in \{0,1\}^{25} \right\}.$$

Es folgt:

ist $\vec{b} \in \text{Bild}(A)$, so ist $\vec{b} \cdot \vec{x} = 0$ für alle $\vec{x} \in \text{Kern}(A)$

Man kann zeigen (1. Semester), dass dies ein "genau dann, wenn" ist.

Also:

$\vec{b} \in \{0,1\}^{25}$ ist lösbar genau dann, wenn $\vec{b} \cdot \vec{x} = 0$ für alle $\vec{x} \in \text{Kern}(A)$

Wir haben oben im Computer berechnet, dass $\text{Kern}(A)$ die Basis

$\vec{y}_1 = (01110\ 10101\ 11011\ 10101\ 01110)$

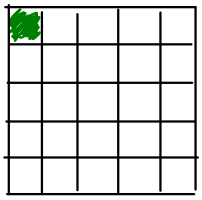
$\vec{y}_2 = (10101\ 10101\ 00000\ 10101\ 10101)$

hat. Wir erhalten damit:

<u>Theorem:</u> $\vec{b} \in \{0,1\}^{25}$ ist lösbar genau dann, wenn

$$\vec{b} \cdot \vec{y}_1 = 0 \text{ und } \vec{b} \cdot \vec{y}_2 = 0.$$

<u>Beispiel:</u>



$\vec{b} = (10000\ 00000\ 00000\ 00000\ 00000)$

$\vec{y}_1 = (01110\ 10101\ 11011\ 10101\ 01110)$

$\vec{b} \cdot \vec{y}_1 = \sum 00000\ 00000\ 00000\ 00000\ 00000 = 0$

$\vec{y}_2 = (10101\ 10101\ 00000\ 10101\ 10101)$

$\vec{b} \cdot \vec{y}_2 = \sum 10000\ 00000\ 00000\ 00000\ 00000 = 1$

$\Rightarrow \vec{b}$ nicht lösbar!