

# **Sudoku-Informatik**

## **oder wie man als Informatiker Logikrätsel löst**

Peter Becker  
Hochschule Bonn-Rhein-Sieg  
Fachbereich Informatik  
`peter.becker@h-brs.de`

Kurzvorlesung am Studieninformationstag, 13.05.2009

---

# Sudoku

Fülle das gesamte Quadrat so aus, dass

- in jeder **Zeile**,
- in jeder **Spalte** und
- in jedem der **Unterquadrate**

die Ziffern 1 bis 9 genau einmal vorkommen.

	9					1	
8		4		2		3	7
	6		9		7		2
		5		3		1	
	7		5		1		3
		3		9		8	
	2		8		5		6
1		7		6		4	9
	3						8

---

## Sudoku: Beispiel

	9					1	
8		4		2		3	7
	6		9		7		2
		5		3		1	
	7		5		1		3
		3		9		8	
	2		8		5		6
1		7		6		4	9
	3						8

7	9	2	3	5	4	6	1	8
8	5	4	1	2	6	3	9	7
3	6	1	9	8	7	5	2	4
9	4	5	6	3	8	1	7	2
2	7	8	5	4	1	9	3	6
6	1	3	7	9	2	8	4	5
4	2	9	8	1	5	7	6	3
1	8	7	2	6	3	4	5	9
5	3	6	4	7	9	2	8	1

---

## Diese Kurzvorlesung soll ...

- typische **Methoden** zeigen, die man im **Informatikstudium** lernt,
- die Bedeutung der **Modellierung** in der Informatik deutlich machen,
- zeigen, was man mit Mathematik und der passenden Software so alles machen kann
- und ein wenig Spaß machen darf sie auch.

---

## Sudoku aus der Sicht des Informatikers

- Die Ziffern dienen nur zur Unterscheidung. Statt der Ziffern könnten auch neun unterschiedliche Symbole verwendet werden.
- Für den Informatiker ist Sudoku ein **Constraint-Satisfaction-Problem (CSP)**:

Die Aufgabe bei einem CSP besteht darin, einen Zustand (d. h. Belegungen von Variablen) zu finden, der alle aufgestellten **Bedingungen (Constraints)** erfüllt.
- Problem der Modellierung: Variablen, Wertebereiche, Bedingungen?

---

## Ein ganz simples CSP

**Variablen:**  $x, y$

**Wertebereiche:**  $D(x) = D(y) = \mathbb{R}$

**Constraints:**

$$2x + 3y = 19$$

$$-x + y = 3$$

---

## CSP-Probleme in der Praxis ...

- haben typischerweise eine deutlich **höhere Zahl an Variablen**,
- weisen **unhandlichere Wertebereiche** auf (z.B.  $\mathbb{N}$ , Boolean, symbolische Werte)
- und besitzen **komplexere Nebenbedingungen** (z.B. Ungleichungen).

---

## Von der Logik zur Algebra (1)

- Sudoku ist ein **Logikrätsel**: Die Lösung erfordert **logische Schlussfolgerungen**.
- Wir überführen die logischen Bedingungen vollständig in ein **algebraisches Modell (Gleichungssystem)**.
- Dann lösen wir das algebraische Modell mit einem mathematischen Verfahren und haben damit auch eine Lösung für das Rätsel.



---

## Von der Logik zur Algebra (2)

**Sprache:** Wenn  $A$  gilt, dann gilt auch  $B$ .

**Logik:**  $A \implies B$

$A$	$B$	$A \implies B$
w	w	w
w	f	f
f	f	w
f	w	w

---

## Von der Logik zur Algebra (3)

**Logische Normalform (KNF):**  $\neg A \vee B$

A	B	$\neg A \vee B$
w	w	w
w	f	f
f	f	w
f	w	w

---

## Von der Logik zur Algebra (4)

**Algebra:**  $A, B \in \{0, 1\}$ :  $(1 - A) + B \geq 1$

A	B	$(1 - A) + B$
1	1	1
1	0	0
0	0	1
0	1	2

---

## Von der Logik zur Algebra (5)

**Logik:**

$$\begin{aligned}(A \wedge \neg B) \Rightarrow (C \vee \neg D) &\equiv \neg(A \wedge \neg B) \vee C \vee \neg D \\ &\equiv \neg A \vee B \vee C \vee \neg D\end{aligned}$$

**Algebra:**

$$(1 - A) + B + C + (1 - D) \geq 1 \text{ für } A, B, C, D \in \{0, 1\}$$

---

## Ein Gleichungssystem für Sudokus (1)

Modellierung mit Hilfe von **booleschen Variablen**:

$$x_{ijk} = \begin{cases} 1 & \text{wenn in Zeile } i \text{ und Spalte } j \text{ Ziffer } k \text{ steht} \\ 0 & \text{sonst} \end{cases}$$

Anzahl an Variablen:

$$\text{Anzahl Zeilen} \times \text{Anzahl Spalten} \times \text{Anzahl Ziffern} = 9^3 = 729$$

---

## Ein Gleichungssystem für Sudokus (2): Eindeutigkeit von Ziffern in Zeilen

**Beispielbedingung:** In Zeile 3 gibt es genau einmal die Ziffer 5:

$$x_{315} + x_{325} + x_{335} + x_{345} + x_{355} + x_{365} + x_{375} + x_{385} + x_{395} = 1$$

**Allgemein:** Für jeder Zeile  $i$  gibt es genau einmal die Ziffer  $k$ :

$$x_{i1k} + x_{i2k} + x_{i3k} + x_{i4k} + x_{i5k} + x_{i6k} + x_{i7k} + x_{i8k} + x_{i9k} = 1$$

Insgesamt 81 Gleichungen

---

## Ein Gleichungssystem für Sudokus (3): Eindeutigkeit von Ziffern in Spalten

**Beispielbedingung:** In Spalte 4 gibt es genau einmal die Ziffer 2:

$$x_{142} + x_{242} + x_{342} + x_{442} + x_{542} + x_{642} + x_{742} + x_{842} + x_{942} = 1$$

**Allgemein:** Für jede Spalte  $j$  gibt es genau einmal die Ziffer  $k$ :

$$x_{1jk} + x_{2jk} + x_{3jk} + x_{4jk} + x_{5jk} + x_{6jk} + x_{7jk} + x_{8jk} + x_{9jk} = 1$$

Insgesamt 81 Gleichungen

---

## Ein Gleichungssystem für Sudokus (4): Eindeutigkeit von Ziffern in Unterquadraten

**Beispielbedingung:** In Unterquadrat links oben gibt es genau einmal die Ziffer 5:

$$x_{115} + x_{125} + x_{135} + x_{215} + x_{225} + x_{235} + x_{315} + x_{325} + x_{335} = 1$$

**Allgemein:** Für jedes Unterquadrat mit linker oberer Ecke in  $i, j$  gibt es genau einmal die Ziffer  $k$ :

$$\begin{aligned} x_{ijk} + x_{i(j+1)k} + x_{i(j+2)k} + x_{(i+1)jk} + x_{(i+1)(j+1)k} + x_{(i+1)(j+2)k} \\ + x_{(i+2)jk} + x_{(i+2)(j+1)k} + x_{(i+2)(j+2)k} = 1 \end{aligned}$$

Insgesamt 81 Gleichungen



---

## Ein Gleichungssystem für Sudokus (5): In jeder Zelle nur eine Ziffer

**Beispielbedingung:** In Zeile 7 und Spalte 1 muss genau eine Ziffer stehen:

$$x_{711} + x_{712} + x_{713} + x_{714} + x_{715} + x_{716} + x_{717} + x_{718} + x_{719} = 1$$

**Allgemein:** Für jede Zeile  $i$  und jede Spalte  $j$  gibt es genau eine Ziffer:

$$x_{ij1} + x_{ij2} + x_{ij3} + x_{ij4} + x_{ij5} + x_{ij6} + x_{ij7} + x_{ij8} + x_{ij9} = 1$$

Insgesamt 81 Gleichungen

---

## Ein Gleichungssystem für Sudokus (6): Vorbelegungen von Zellen mit Ziffern

**Beispielbedingung:** In Zeile 1 und Spalte 2 steht eine 9:

$$x_{129} = 1$$

Insgesamt so viele Gleichungen wie Vorbelegungen

---

## Berechnung der Lösung

- Vollständige Modellierung als Gleichungssystem mit booleschen Variablen
- Einsatz eines Gleichungslösers aus dem Gebiet der Linearen Programmierung: [GNU Linear Programming Toolkit](#)
- Eigentlich für die Lösung [linearer Optimierungsprobleme unter Nebenbedingungen](#) gedacht, hier: Dummy-Zielfunktion

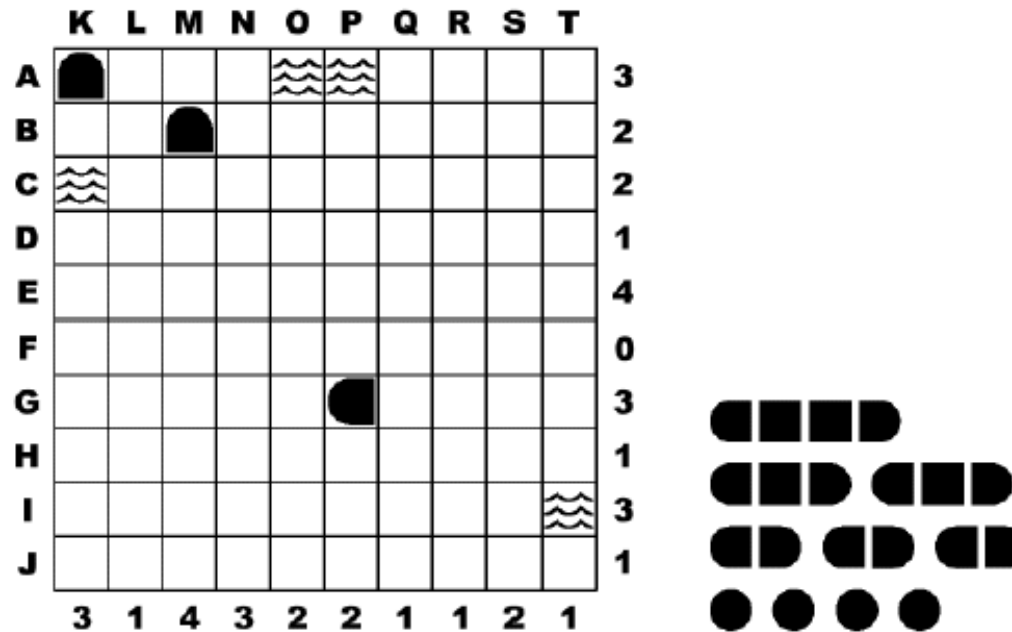
 Demonstration

---

## Ergebnisse

- Selbst schwere Sudoku-Probleme in Sekundenbruchteilen lösbar
- Keine Effizienzunterschiede zu dedizierten Sudokulösern erkennbar
- Kann auch zur Eindeutigkeitsprüfung und zur Generierung von Problemen eingesetzt werden

## Andere Logikrätsel: Battleship Puzzle



Wo liegt die Flotte? Schiffe berühren sich nicht, auch nicht diagonal. Die Zahlen geben die Schiffsteile pro Zeile bzw. Spalte an.

---

## Andere Logikrätsel: Battleship Puzzle (2)

Variablen:

$$x_{ijlr} = \begin{cases} 1 & \text{wenn in Zeile } i \text{ und Spalte } j \text{ ein Schiff der Länge } l \\ & \text{mit Ausrichtung } r \text{ beginnt} \\ 0 & \text{sonst} \end{cases}$$

- Ausrichtung: **horizontal** oder **vertikal**
- $x_{ijlr}$  beschreibt linke (horizontal) bzw. obere (vertikal) Ecke eines Schiffs.

---

## Andere Logikrätsel: Battleship Puzzle (3)

- Anzahl der Variablen deutlich größer als bei Sudoku:

$$\approx \text{Zeilen} \times \text{Spalten} \times \text{Längen} \times 2$$

- plus Hilfsvariablen  $y_{ij}$  die angeben, ob in Zeile  $i$  und Spalte  $j$  ein Schiffsteil liegt.
- Variablen durch Ungleichungen verbunden, z.B.

$$x_{ak2v} - y_{ak} \leq 0$$

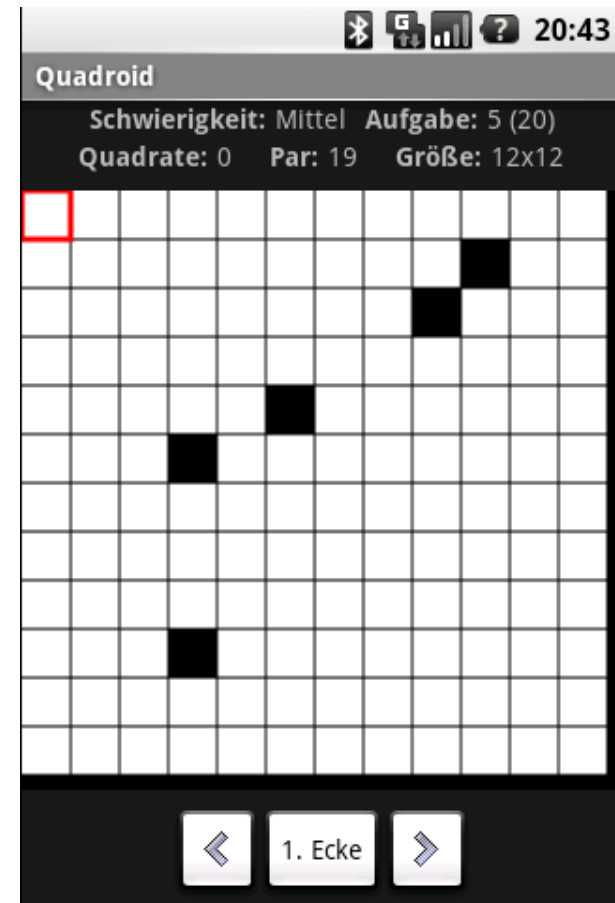
$$x_{ak2v} + y_{al} \leq 1$$

- 
- Für Probleme der Größe 12x12 ca. 20.000 Ungleichungen
  - Lösungszeiten: bis zu mehreren Minuten (bei 12x12-Probleme)



## Andere Logikrätsel: Quadroid

- Fülle das große Quadrat mit möglichst wenigen kleinen Quadraten aus!
- Die schwarzen Blöcke können nicht überdeckt werden.
- **Optimierrätsel**, programmiert für Android-Handys (z.B. T-Mobile G1)
- Berechnung der “par”-Werte für das Rätsel?

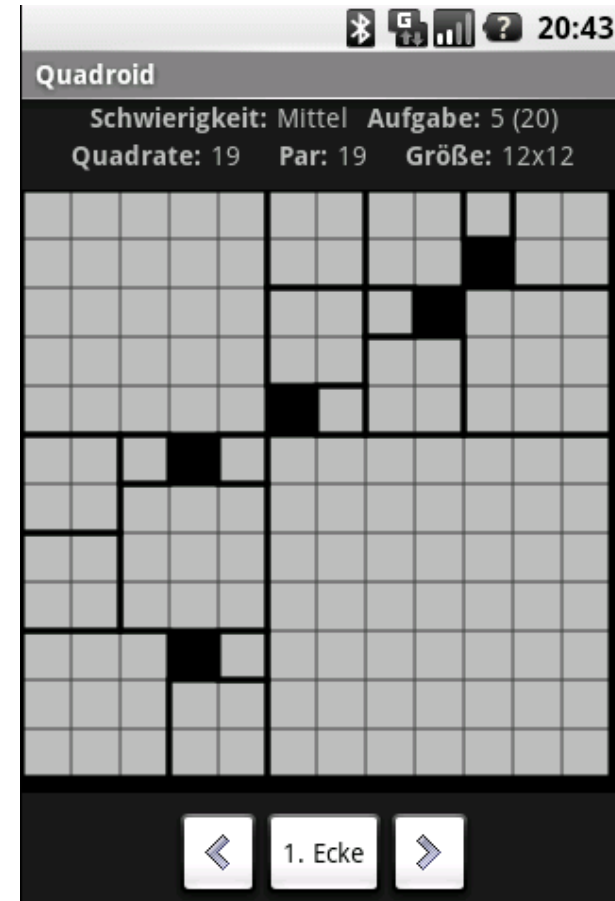


## Andere Logikrätsel: Quadroid (2)

- $x_{ijs}$ : In Zeile  $i$  und Spalte  $k$  liegt linke obere Ecke eines Quadrats der Größe  $s$
- Beispielbedingung:

$$x_{111} + x_{112} + x_{113} + x_{114} + x_{115} \geq 1$$

- Für Problem rechts **365 Variablen** und **139 Nebenbedingungen**
- **Berechnungszeit**: im Sekundenbereich



---

# Zusammenfassung

- **Abstraktion** und **Modellbildung** sind wichtige Komponenten in der Informatik und im Informatikstudium.
- **Informatiker sind Problemlöser!**  
Analysieren, Modellieren, Entwerfen, Implementieren, Testen
- Hier: **Algebraische Modellierung**, ermöglicht den Einsatz von Standardsoftware zur einfachen “Problemlösung”

---

Vielen Dank für Ihre Aufmerksamkeit!

