

**TIM**tutor  
OpenSource Project

HUMBOLDT-UNIVERSITÄT ZU BERLIN



# TIM-Tutor - Theoretische Informatik mit *Mathematica* und *webMathematica*

*Klaus-Peter Neuendorf, Frank Hambach, Tino Naphtali, Hans Schemmel*

**Lernplattformen im Internet werden in Zukunft eine wachsende Rolle bei der Realisierung lebenslangen und selbstgesteuerten Lernens in allen Bereichen der Bildung und Weiterbildung von Jugendlichen und Erwachsenen spielen.**

**Games are becoming ever more popular as tools for learning, both in teaching and the workplaces. Despite major differences of opinion among different disciplines, both researchers and developers agree on one thing: The strength of games is that you learn things in the context in which you use them.**

*Susanne Ekman*

**Beispiel: Determinanten - Tic-Tac-Toe**

## Die Vision

**TIM – Tutor** (steht für **Theoretische Informatik mit Mathematica – Tutor**)

wird im **Dialog** mit den **Studenten der Informatik**

- **Definition und Bedeutung von Begriffen klären**
- **(Bei) Spiele und Aufgaben präsentieren sowie**
- **Arbeitshinweise zur Lösung von Übungsaufgaben und auch**
- **zur Vorbereitung auf Prüfungen geben.**

**TIM – Tutor** wird als **AIML – basierter Webbot**

(**AIML – Artificial Intelligence Markup Language**)

- **didaktisch fundierte Gegenfragen und**
- **Hinweise auf konkrete Anfragen der Studenten geben, sowie**
- **interaktiv Definitionen, Beispiele, Übungsaufgaben und**
- **Musterlösungen Online mit Hilfe von Mathematica und webMathematica Funktionen präsentieren.**

The screenshot shows the TIM-Tutor web interface. The browser window title is "TIM-Tutor: Elementare Mengenlehre - Mozilla". The address bar contains "http://www.informatik.hu-berlin.de/~neuendor/prosem04/". The page content is titled "Elementare Operationen mit Mengen". On the left, there is a sidebar with a "Thematik" menu listing: VEREINIGUNG, DURCHSCHNITT, DIFFERENZ, SYMMETRISCHE DIFFERENZ, KOMPLEMENT, POTENZMENGE, and Mathematica HINTERGRUND. The main content area shows a task: "Berechne die Vereinigung:  $A \cup B = \{7, 11, 14, 16\} \cup \{5, 7, 8, 14\}$ ". Below this is an input field containing "{ }" and an "Evaluate" button. The result shown is "0". To the right, there is a "Venn-Diagramm Notation" section with three diagrams: "A =", "B =", and "A U B =". The "A U B =" diagram shows two overlapping circles labeled "A" and "B". At the bottom, there is a "Kommentar" field with the text "Das war |False| , nämlich gleich 0" and a "Student:" field with the name "Weiter".

## Das Projekt

wird gefördert  
durch das **Multimedia Programm 2003**  
der **Humboldt – Universität zu Berlin**

## Die Teilnehmer

**Dr. Klaus – Peter Neuendorf**,  
(ehemaliger Mitarbeiter des Instituts für Informatik der HU)

**Frank Hambach** (Student Informatik, wissenschaftliche Hilfskraft)

**Tino Naphtali** (Student 8. Semester Informatik)

**Dr. Hans Schemmel** (Diplom – Mathematiker im Ruhestand)

sowie weitere Studenten des Seminars zum Thema E –  
Learning im SS' 03 am Institut für Informatik der Humboldt – Universität

## Das Programm

**I. Entwicklung von *Mathematica* Notebooks zur  
Realisierung der Datenstrukturen und Algorithmen  
der Theoretischen Informatik in folgenden Bereichen**

### **MENGENLEHRE**

- Mengenalgebra
- Korrespondenzen und Relationen
- Funktionen
- Induktive Definitionen

### **WÖRTER, SPRACHEN und GRAMMATIKEN**

### **BOOLESCHE FUNKTIONEN**

### **AUSSAGENKALKÜL**

### **BERECHENBARKEIT**

- Partiiell Rekursive Funktionen
- Turingmaschinen

**II. Übertragung der Funktionalität der Notebooks  
auf Webpages mit Hilfe von *webMathematica***

**III. Realisierung einer Lernplattform  
(einheitlichen Oberfläche) mit dem Webbot *TIM – Tutor***

## Die Highlights (so far)

### ■ Mengenlehre mit *Mathematica*

Obwohl Mengen als Datenstruktur in *Mathematica* nicht direkt unterstützt werden, gibt es eine Reihe von Autoren, die sich ebenfalls damit beschäftigt haben.

```
DISCRETE MATHEMATICS 1.0 by  
Geoff.H.Smith@uts.edu.au  
Nicolae.Nicorovici@uts.edu.au,  
MathSource 0212 - 230,  
2002 - 08 - 09
```

```
On Relations by  
Jaime Rangel - Mondragon,  
MathSource 0212 - 083, 2002 - 05 - 01
```

```
Mengen von  
Rolf Sulanke,  
vorgestellt auf seiner Webseite , 2001
```

Im TIM-Tutor Projekt wird eine Sammlung von *Mathematica* Notebooks entwickelt, in denen der Übungsstoff neben der Darstellung und Erläuterung der Datenstrukturen auch durch randomisierte Aufgabenstellungen mit Lösungsüberprüfung vermittelt wird.

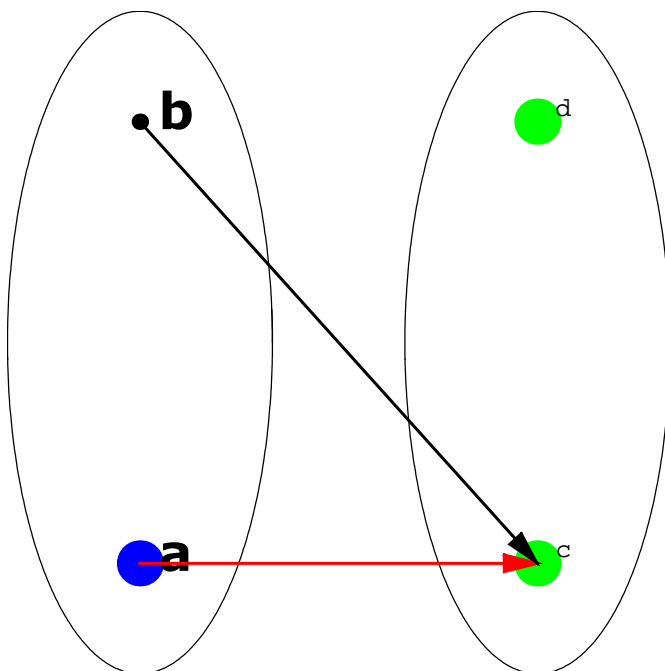
```
TIMchapter1 Modul : Mengen  
TIMchapter2 Modul : Korrespondenzen und Relationen  
TIMchapter3 Modul : Abbildungen  
TIMchapter4 Modul : Wörter und Sprachen  
TIMchapter5 Modul : Induktive Definitionen &  
Beweise  
TIMchapter6 Modul : Boolesche Funktionen  
TIMchapter7 Modul : Aussagenkalkül
```

## ■ Datenstrukturen und Visualisierungen von Korrespondenzen und Relationen

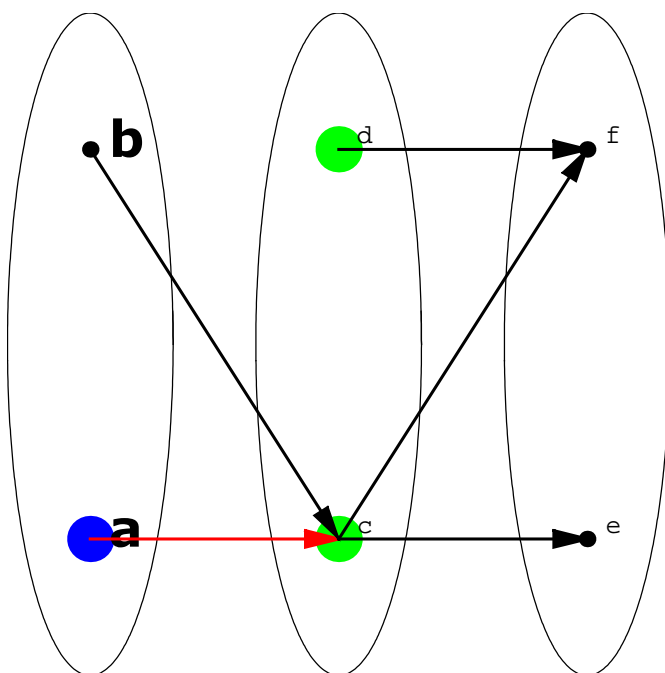
vorgestellt und entwickelt von *Frank Hambach*

Der Zweck dieses Packages ist, es ein möglichst einfaches Interface zum Erzeugen von Darstellungen wie den folgenden für Korrespondenzen und Relationen bereitzustellen. Dabei wurde versucht der Philosophie des Combinatorica-Packages für Datenstrukturen und Algorithmen von Steven Skiena und Sriram Pemmaraju zu folgen.

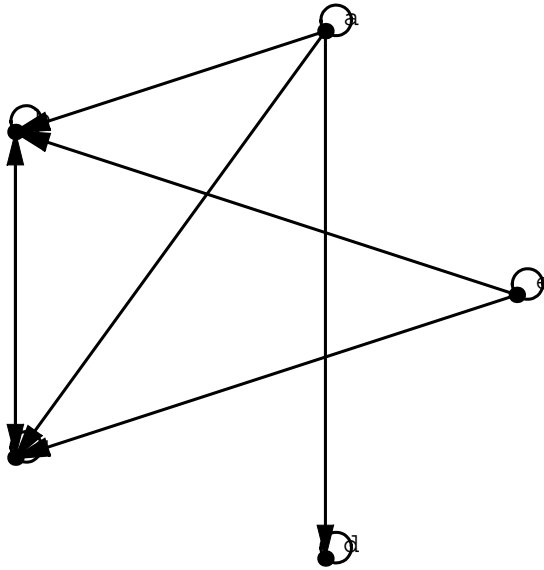
```
ShowCorrespondenceGraph[cor1];
```



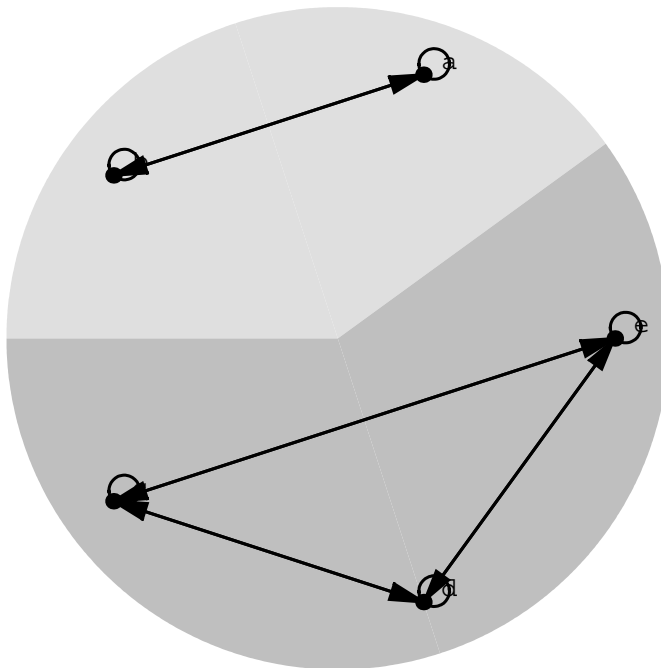
```
ShowCorrespondenceGraph[cor1, cor2];
```



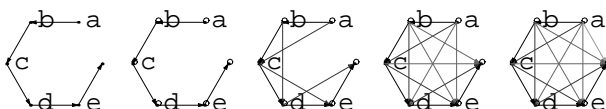
```
ShowRelationGraph[ReflexiveTransitiveClosure[rel1]];
```



```
ShowEquivalenceClasses[
  PairsToRelation[{{a, b}, {b, a}, {a, a}, {b, b}, {c, c}, {d, d}, {e, e},
    {d, e}, {e, d}, {c, d}, {d, c}, {e, c}, {c, e}}, {a, b, c, d, e}];
```



```
Show[
  GraphicsArray[ShowReflexiveTransitiveClosure[PairsToRelation[{{a, b},
    {b, c}, {c, d}, {d, e}, {e, f}}, {a, b, c, d, e, f}], Graphics]]];
```



Das Package wird eingebunden in Webseiten wie den folgenden des TIM-Tutor-Projektes

1. Der **Definitionsbereich** (Präimagebereich, Urbereich) von  $K$  ist die Menge  $D_K$  der Elemente von  $M$ , die  $K$ -Urbilder eines Elementes von  $N$  sind, d.h.  $D_K = \{a \mid \text{es gibt ein } b \text{ mit } [a, b] \in K\}$ .

2. Der **Bildbereich** (Wertebereich, Nachbereich) von  $K$  ist die Menge  $B_K$  der Elemente von  $N$ , die  $K$ -Bild eines Elementes von  $M$  sind, d.h.  $B_K = \{b \mid \text{es gibt ein } a \text{ mit } [a, b] \in K\}$ .

3.  $K$  heißt **Korrespondenz von  $M$  in  $N$** , wenn  $D_K = M$  ist.

4.  $K$  heißt **Korrespondenz aus  $M$  auf  $N$** , wenn  $B_K = N$  ist.

5.  $K$  heißt **Korrespondenz aus  $M$  auf  $N$** , wenn  $D_K = M$  ist und  $B_K = N$  ist.

6. Die Korrespondenz  $K$  wird **eindeutig** genannt, wenn zu jedem  $a \in M$  höchstens ein  $b \in N$  (eventuell keines) mit  $[a, b] \in K$  existiert.

7. Die Korrespondenz  $K$  wird **eindeutig umkehrbar** genannt, wenn es zu jedem  $b \in N$  höchstens ein  $a \in M$  mit  $[a, b] \in K$  gibt.

Definition ausprobieren

Korrespondenz

aus der Menge

in die Menge

**Definitions- und Wertebereich anzeigen**

Definitionsbereich

```
graph LR; subgraph M; a; b; end; subgraph N; e; f; g; end; a --> e; a --> f; b --> e; b --> g;
```

## ■ Anzahlaufgaben mit Venn-Diagrammen

Der vorliegende Aufsatz beschreibt *Mathematica* Programme zum Lösen und zum Generieren von randomisierten Aufgaben zur Mengenlehre vom folgenden Typ.

### Musteraufgabe\*

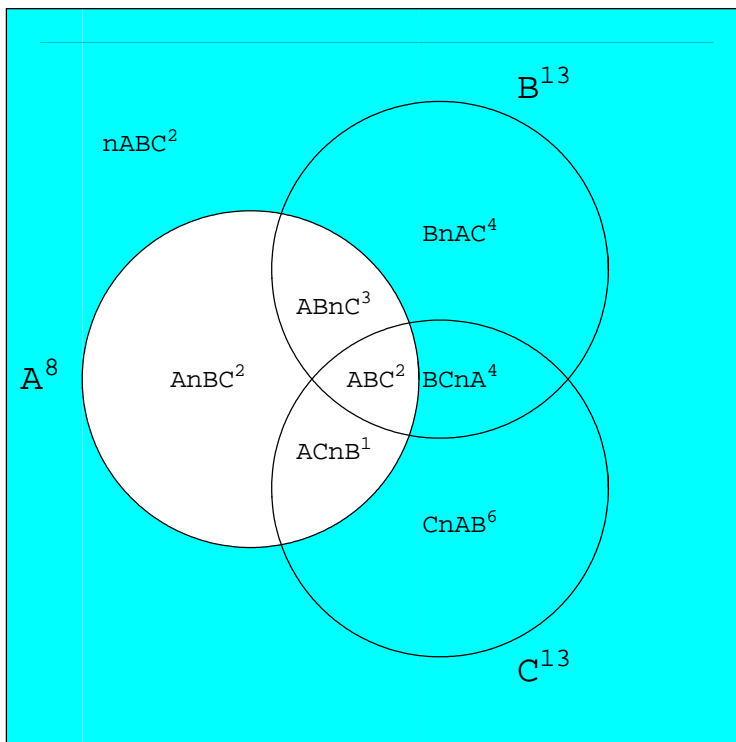
Ein Professor hat zwei Dutzend (ein Dutzend=12) Informatikbücher, die sich u.a. mit den Gebieten (**A**) Compilerbau, (**B**) Datenbanken und (**C**) Betriebssystemen beschäftigen. Folgende Information über die Anzahl der Bücher und die von ihnen behandelten Themen ist bekannt:

$$|A| = 8; |B| = 13; |C| = 13; |A \cap B| = 5; |A \cap C| = 3; |B \cap C| = 6; |A \cap B \cap C| = 2$$

- (a) Wieviel der Bücher enthalten Material aus genau einem der genannten Gebiete?  
 (b) Wieviel der Bücher beschäftigen sich weder mit Compilerbau noch mit Datenbanken oder mit Betriebssystemen?  
 (c) Wieviel der Bücher beinhalten kein Material über Compilerbau?

Quelle: Übungsserie 1 zur Vorlesung Theoretische Informatik I im WS'03 am Institut für Informatik der Humboldt-Universität

Es wird immer von einer konkreten Allmenge ausgegangen, welche alle betrachteten Elemente umfaßt und im Programm mit `allSet` bezeichnet wird. Wird diese nicht explizit angegeben, wird die Vereinigung aller betrachteten Mengen als Allmenge angenommen. Dann legen wir die Anzahl  $n$  der beteiligten Grundmengen fest. Hierbei gehören 2, 3, und eventuell noch 4 zu den Standardmengenanzahlen. Die Zahl von  $3^n$  Variablen und  $3^n - 2^n$  Gleichungen die im Verlauf unserer Lösung betrachtet werden, zeigt den exponentiellen Charakter der Aufgabe, bzw. unseres Algorithmus, der damit im allgemeinen schon ab  $n=5$  per Hand und ab  $n=9$  auch per Computer sehr schwierig zu realisieren ist. In obiger Aufgabe ist offensichtlich  $n=3$ . Unser Programm ermittelt für diese Aufgabe folgende im Venn-Diagramm repräsentierte Lösung. Die Bezeichner kodieren einen mengenalgebraischen Ausdruck und die Exponenten die entsprechenden Elementanzahlen.





$$\begin{aligned} \bar{A} \cap B \cap C \cup \bar{A} \cap B \cap \bar{C} \cup A \cap B \cap C &= 12 \\ \bar{A} \cap \bar{B} \cap \bar{C} &= 2 \\ \bar{A} &= 16 \end{aligned}$$

Unser *Mathematica* Programm ermöglicht die randomisierte Erzeugung von Aufgabenstellungen dieser Art. Dabei werden auch Fälle in denen die Lösung nicht eindeutig ist, wie im nachfolgenden Beispiel berücksichtigt.

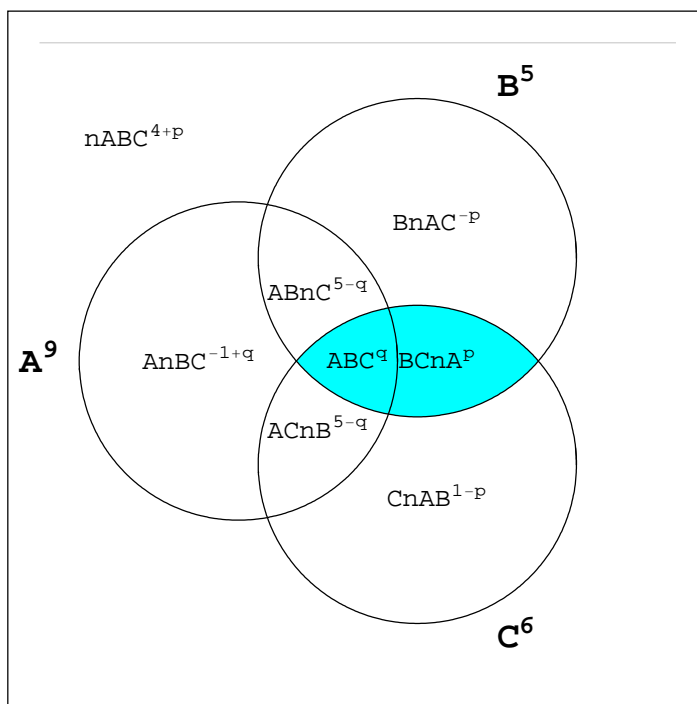
## Beispiel mit endlich vielen verschiedenen Lösungen

```
ClearAll[solution, wVs, cRs, showVenn];
SeedRandom[1];
randomExample[3, 4, solution, wVs, cRs]
```

```
{{|C| = 8, |B| = 5, |A ∩ B| = 5,
  |A ∩ C| = 4, |A̅| = 5, |A ∩ C| = 5, |A ∩ C̅| = 4, |A̅ ∩ B| = 0},
{|A ∩ B ∩ C|, |A ∩ B ∩ C̅|, |A ∩ B|, |A ∩ B̅ ∩ C|, |A ∩ B̅ ∩ C̅|, |A ∩ B̅|,
|A|, |A̅ ∩ B ∩ C|, |A̅ ∩ B ∩ C̅|, |A̅ ∩ B̅ ∩ C|, |A̅ ∩ B̅ ∩ C̅|, |A̅ ∩ C|,
|B ∩ C|, |B ∩ C̅|, |B̅ ∩ C|, |B̅ ∩ C̅|, |B|, |C|, |allSet|}}
```

Im obigen stellt die erste Menge die gegebenen Größen dar, während die zweite Menge alle nicht gegebenen Größen darstellt. Es wird die allgemeine Lösung berechnet und für das Venn-Diagramm kann die anzuzeigende Teilmenge gewählt werden. Die Bezeichnung  $v[u, \text{True}, \text{True}]$  steht für den Durchschnitt von B und C.

```
toSolveInput[3, {v[u, True, True]}, cRs, showVenn → True]
```



$$|B \cap C| = p + q$$

`toSolveInput[3, wVs, cRs]`

$$\begin{aligned} |A \cap B \cap C| &= q \\ |A \cap B \cap \bar{C}| &= 5 - q \\ |A \cap B| &= 5 \\ |A \cap \bar{B} \cap C| &= 5 - q \\ |A \cap \bar{B} \cap \bar{C}| &= -1 + q \\ |A \cap \bar{B}| &= 4 \\ |A| &= 9 \\ |\bar{A} \cap B \cap C| &= p \\ |\bar{A} \cap B \cap \bar{C}| &= -p \\ |\bar{A} \cap \bar{B} \cap C| &= 1 - p \\ |\bar{A} \cap \bar{B} \cap \bar{C}| &= 4 + p \\ |\bar{A} \cap C| &= 1 \\ |B \cap C| &= p + q \\ |B \cap \bar{C}| &= 5 - p - q \\ |\bar{B} \cap C| &= 6 - p - q \\ |\bar{B} \cap \bar{C}| &= 3 + p + q \\ |\bar{B}| &= 9 \\ |C| &= 6 \\ |\text{allSet}| &= 14 \end{aligned}$$

Die Startlösung von der ausgehend die Aufgabenstellung erzeugt wurde, um sicher zu gehen, daß wenigstens eine Lösung existiert, war die folgende.

**solution**

$$\{ |A \cap B \cap C| = 3, |A \cap B \cap \bar{C}| = 2, |A \cap B| = 5, |A \cap \bar{B} \cap C| = 2, \\ |A \cap \bar{B} \cap \bar{C}| = 2, |A \cap \bar{B}| = 4, |A| = 9, |\bar{A} \cap B \cap C| = 0, |\bar{A} \cap B \cap \bar{C}| = 0, \\ |\bar{A} \cap \bar{B} \cap C| = 1, |\bar{A} \cap \bar{B} \cap \bar{C}| = 4, |\bar{A} \cap C| = 1, |B \cap C| = 3, |B \cap \bar{C}| = 2, \\ |\bar{B} \cap C| = 3, |\bar{B} \cap \bar{C}| = 6, |\bar{B}| = 9, |C| = 6, |\text{allSet}| = 14 \}$$

## ■ Logik mit Minesweeper

Motivationsquelle Vorlesungsskript von Martin Grohe, <http://www.informatik.hu-berlin.de/logik/lehre/logik/kap1.pdf>

## ■ Spielbares Minesweeper: Train your brain

Change Board size    Change mines number

Mines : 9

more mines    less mines

New Game    Start Game

Hidden free fields : 17

			1		1	
		1	2		2	1
1	1	2	*			
	*	3				
					1	1
					1	
1					1	



Zum Markieren von Minen den Button aktivieren und danach wieder deaktivieren!

**USE LOGIC**

Zum Logik-Modell siehe <http://www.informatik.hu-berlin.de/logik/lehre/logik/kap3.pdf>

Das Spielfeld erst mal ganz klein :-), Felder können vermint sein (**True**) oder frei (**False**).

```
n = m = 3;
gameBoard = Table[True | False, {n}, {m}];
```

Das sind jetzt einfach die Koordinaten des Spielfelds, links oben mit  $\{1, 1\}$  beginnend und ein Pattern für mögliche Positionen.

```
gameBoardPositions = Position[gameBoard, Verbatim[True | False]]
```

```
{ {1, 1}, {1, 2}, {1, 3}, {2, 1}, {2, 2}, {2, 3}, {3, 1}, {3, 2}, {3, 3} }
```

```
gameBoardPositionPattern = Alternatives @@ gameBoardPositions
```

```
{1, 1} | {1, 2} | {1, 3} | {2, 1} | {2, 2} | {2, 3} | {3, 1} | {3, 2} | {3, 3}
```

Die Minenanzahl auf den Nachbarfeldern eines Feldes kann maximal 8 sein und -1 falls noch unbekannt.

```
minesNbPattern = Alternatives @@ Range[-1, 8]
```

```
-1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8
```

Die allgemeine Nachbarumgebung sieht **nbTable** wird dann für das Feld  $\{i, j\}$  als **nb** $\{\{i, j\}$  auf das Spielfeld eingeschränkt.

```
nbTable =  
  {{-1, 1}, {0, 1}, {1, 1}, {1, 0}, {1, -1}, {0, -1}, {-1, -1}, {-1, 0}};
```

```
nb[{i_, j_}] := (Plus[{i, j}, #] & /@ nbTable) ∩ gameBoardPositions
```

Nachfolgend zwei charakteristische Beispiele

```
nb[{1, 1}]
```

```
{ {1, 2}, {2, 1}, {2, 2} }
```

```
nb[{2, 2}]
```

```
{ {1, 1}, {1, 2}, {1, 3}, {2, 1}, {2, 3}, {3, 1}, {3, 2}, {3, 3} }
```

Die Nachbarpattern beschreiben nun alle möglichen Kombinationen von Umgebungen.

```
nbPattern[q : {i_, j_}] := Alternatives @@ Delete[Subsets@nb[q], 1]
```

```
nbPattern[{2, 2}]
```

Warum wird die Ausgabe hier wohl nicht angezeigt?

```
nbPattern[{2, 2}] // Dimensions
```

```
{255}
```

Alles klar?!

```
nbPattern[{1, 1}] // Dimensions
```

```
{7}
```

```
nbPattern[{1, 1}]
```

```
{{2, 2}} | {{2, 1}, {2, 2}} | {{2, 1}} | {{1, 2}, {2, 1}} |
  {{1, 2}, {2, 1}, {2, 2}} | {{1, 2}, {2, 2}} | {{1, 2}}
```

```
nbPattern[{1, 2}] // Dimensions
```

```
{31}
```

Das waren wieder die typischen Vertreter. Jetzt werden die logischen Variablen eingeführt.

```
var[x_, q : gameBoardPositionPattern, s : minesNbPattern] :=
  Subsuperscript[x, q, s]
```

```
var[x, {1, 1}, #] & /@minesNbPattern
```

```
 $x_{\{1,1\}}^{-1} \mid x_{\{1,1\}}^0 \mid x_{\{1,1\}}^1 \mid x_{\{1,1\}}^2 \mid x_{\{1,1\}}^3 \mid x_{\{1,1\}}^4 \mid x_{\{1,1\}}^5 \mid x_{\{1,1\}}^6 \mid x_{\{1,1\}}^7 \mid x_{\{1,1\}}^8$ 
```

Die Semantik ist dabei so, daß der Fußindex das Feld und der Exponent die Zahl der Nachbarminen bzw bei -1, die eigenen Vermintung bzw. Freiheit anzeigt. Kann  $x_{\{1,1\}}^{-1}$  aus allen bisher erkannten Formeln gefolgert werden, ist das Feld nicht vermint.

Jetzt folgen im Skript 3 Typen von aussagenlogischen Formeln, von uns minimal ergänzt.

**phi:** Keine zwei verschiedenen Variablen eines Feldes können gleichzeitig wahr sein, also höchstens eine kann wahr sein.

**psi:** Hiermit soll ausgedrückt werden: Wenn bekannt ist, daß eine gewisse Teilumgebung **nbs** von **q** komplett vermint ist, dann ist die Anzahl aller Minen in der Umgebung von **q** mindestens genauso hoch wie die Anzahl der Felder dieser Teilumgebung. (Aber es kann auch **q** selbst vermint sein!)

**Aus dem Spezialfall leere Umgebung in psi folgt:** Wenigstens eine der Variablen für jedes Feld **q** muß wahr sein.

**chi:** Wenn in der Umgebung von **q** genau **s** Minen sind, dann gibt es mindestens eine Teilmenge der Umgebung von **q** mit **s** Elementen (Feldern), die alle vermint sind.

## ■ Bevor wir alle diese Formeln wirklich erzeugen, zählen wir doch mal wieviele wir für unser

### Mini-Beispiel n=3 erhalten :-)

Hier schon etwas optimiert und nicht rein nach Vorlesungsmodell.

(1) **Erstens:** Anzahl der Variablen:

**Variante (A):** die tricky automatische Erzeugung aller Variablen

```
var[x, {#1, #2}, #3] &@@@ Partition[
  (Outer[List, {#1}, List @@ #2[[Range[1, 2 + (nb[#1] // Length)]]], 1] &[#,
    minesNbPattern] & /@ gameBoardPositions) // Flatten, 3] // Length
```

58

**Variante (B):** die Überprüfung mit Köpfchen

1 Feld hat 8 Nachbarn und je 4 Felder haben 3 bzw 5 Nachbarfelder. Also gibt es  $10 + 4(5 + 7)$  Variablen.

$10 + 4(5 + 7)$

58

(2) **Zweitens:** Wieviele Formeln vom Typ **phi** gibt es?

Na, zu jedem Feld wird jedes Paarmenge von verschiedenen Variablen betrachtet, also brauchen wir nur die berühmte Gaußsche Summenformel oder, *Mathematica* benutzt die sicher sowieso, gleich direkt:

```
Sum[j, {j, #}] & /@ {10, 7, 5}
```

{55, 28, 15}

Also 227 Formeln.

$55 + 4(28 + 15)$

227

(3) **Drittens:** **psi**

Hier brauchen wir wieder die Anzahl der verschiedenen Nachbarschaftspattern von oben.

$255 + 4(31 + 7)$

407

(4) **Viertens:** **chi**

$8 + 4(3 + 5)$

40

Also summa summarum 58 Variablen und  $227+407+40$  Formeln oder Ausdrücke von denen einige recht länglich sind. Zwei Beispiele folgen.



## ■ Die Schwierigkeiten der Umsetzung

Für ein 3x3 Spielfeld erhält man also summa summarum 58 Variablen und 674 Formeln oder Ausdrücke von denen einige recht länglich sind (bis zu 70 Glieder). Für ein einziges Feld, welches nicht am Rand liegt, kommen dann noch 10 Variablen und  $55+255+8$ , also 318 Formeln dazu, so daß sich also schon für die Anfängerstufe in der Microsoft Version ein System mit über zwanzigtausend Formeln und über fünfhundert Variablen ergibt. Warum kann ein Mensch dieses komplexe System auch im Kopf beherrschen?

Die Antwort ist, weil Entscheidungen lokal getroffen werden können.

Die Fortsetzung dieser Arbeit beschreibt die Algorithmen, welche das logische Schließen im modellierten System ermöglichen und soll Studenten zur ihrer bewußten Anwendung motivieren.

---

## Probleme und Wünsche an web $Mathematica$ bzw. die Community

- Automatische Konvertierung von Buttons wie den obigen aus  $Mathematica$  Notebooks zu web $Mathematica$  JSP-Seiten