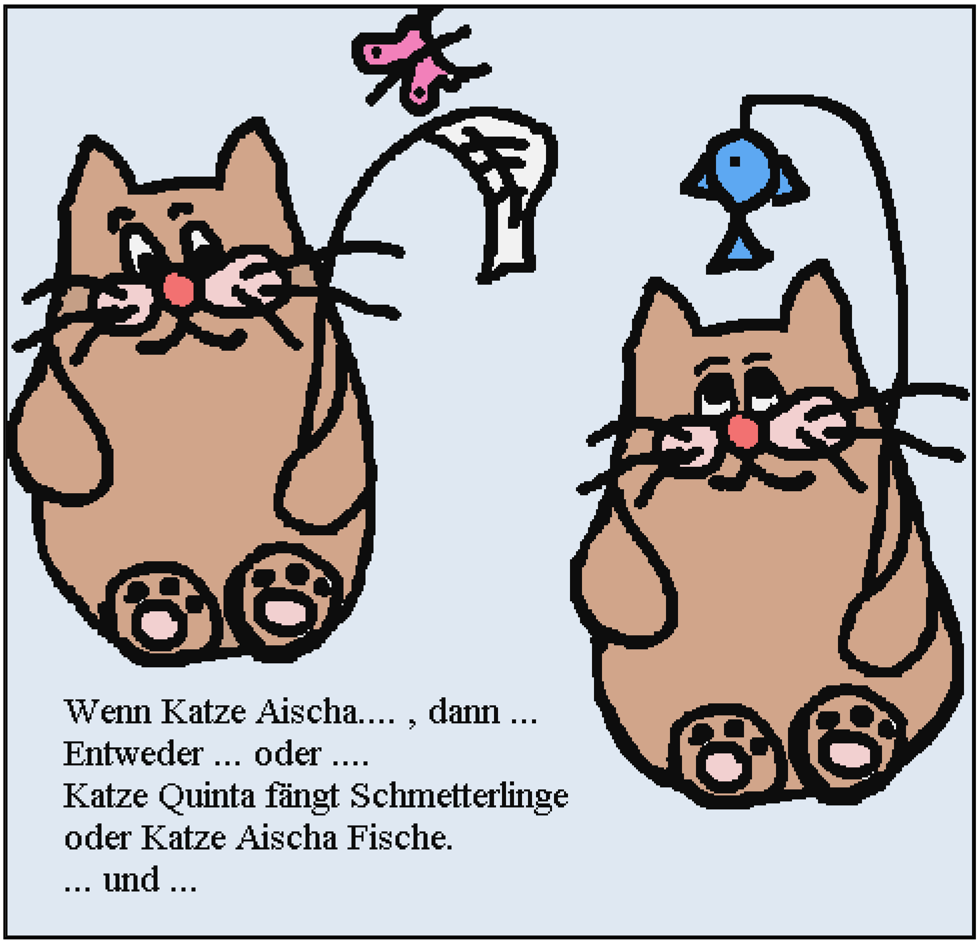
Hilfsmittel zum Lösen und Aufstellen von Logeleien

Inna Tishchenko



Seeland Gymnasium Biel, Deutsches Gymnasium
Maturjahrgang 2007
Klasse 1f

Betreuung: Daniel Diserens

Inhaltsverzeichnis

1. Einleitung

1.1. Vorwort oder es braucht kein logisches Denken um Logeleien zu lösen

> *„Die Herren Amann, Bemann, Cemann und Demann heissen – nicht unbedingt in derselben Reihenfolge – mit Vornamen Erich, Fritz, Gustav und Heinrich. Sie sind alle verheiratet. Ausserdem weiss man über sie und ihre Ehefrauen noch dies: Entweder ist Amanns Vorname Heinrich, oder Bemanns Frau heisst Inge. Wenn Cemann mit Josefa verheiratet ist, dann – und nur in diesem Falle – heisst Klaras Mann nicht Fritz.*
> *Wenn Josefas Mann nicht Erich heisst, dann ist Inge mit Fritz verheiratet.*
> *Wenn Luises Mann Fritz heisst, dann ist der Vorname von Klaras Mann nicht Gustav.*
> *Wenn die Frau Von Fritz Inge heisst, dann ist Erich nicht mit Josefa verheiratet.*
> *Wenn Fritz nicht mit Luise verheiratet ist, dann heisst Gustavs Frau Klara.*
> *Entweder ist Demann mit Luise verheiratet, oder Cemann heisst Gustav.*
> *Wie heissen die Herren mit vollem Namen, wie ihre Ehefrauen mit Rufnamen?“ (Zweistein, 99 Logeleien von Zweistein, 1974, Aufgabe 94)*

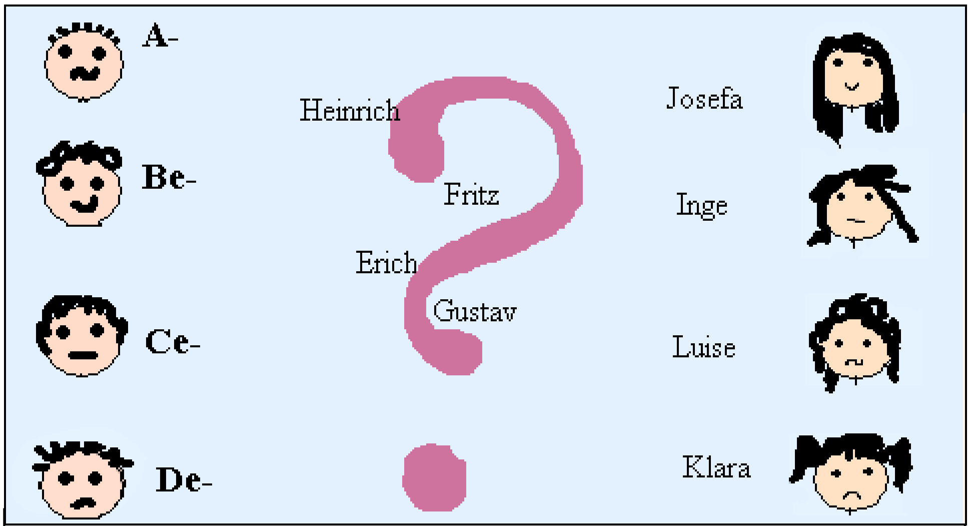


Abbildung der Aufgabe 94

Ist Cemann mit Josefa verheiratet? Heisst Demann Gustav?
Wie löst man solche Aufgaben? Muss man gut logisch denken können, um Logeleien zu lösen? Ist es ein Zufall, dass diese Aufgabe nur eine Lösung hat, oder steckt etwas mehr dahinter? Kann ich auch eine Logelei aufstellen?
Mathematik ist seit sehr langem mein Lieblingsfach, und da ich besonders gern Aufgaben habe, die logisches Denken fordern, habe ich mich entschlossen, meine Maturarbeit dem Thema „Logeleien“ zu widmen.
Es macht mir Spass, zu forschen und herauszufinden, wie etwas funktioniert, denn dieses Vorgehen bietet unter anderem die Möglichkeit, etwas Neues zu entdecken. So habe ich mich auch entschieden, nicht nur den praktischen Teil, sondern auch die ganze Theorie, Lösungs- und Aufbaumethoden für meine Maturarbeit selbständig zu entwerfen und zu erarbeiten. Das einzige Buch, auf das ich mich gestützt habe, ist „99 Logeleien von Zweistein“ mit den Aufgabestellungen. Ich hoffe, dass es mir gelungen ist, etwas entdeckt und erarbeitet zu haben, was noch nicht erforscht worden ist.

Das Ziel meiner Arbeit war somit, *selbständig* ein Hilfsmittel zu erarbeiten, das den LeserInnen zur Verfügung stehen würde, um Logeleien zu *verstehen*, zu *lösen* und selbständig *aufzubauen*.
Diese Arbeit zeigt unter anderem, dass *jeder* Logeleien lösen kann. Es braucht kein logisches Denken, um Logeleien lösen zu *können*.

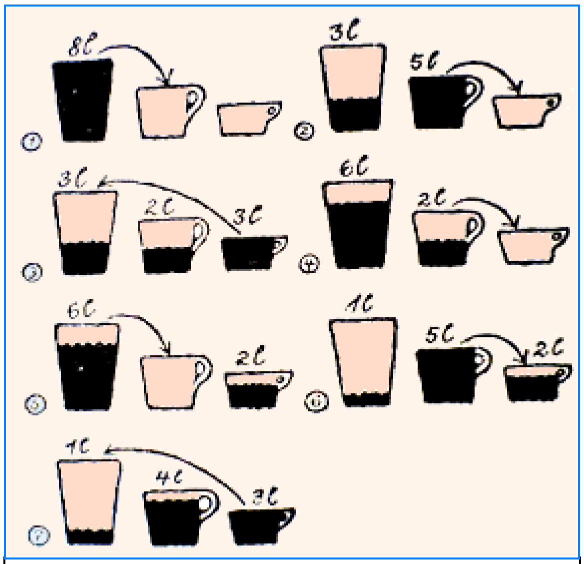
1.2. Einleitung oder was sind Logeleien?

Logeleien, oder auch Logicals genannt, sind interessante mathematische Aufgabestellungen, die uns fordern, **logisch** zu denken.

Es gibt verschiedene Arten von Logeleien, und ich habe mich mit der Art der **Textaufgaben** beschäftigt, die Aussagen enthalten und die aus den logischen Relationen *„Gleich“, „Ungleich“, „Grösser“, „Grösser oder gleich“, „Kleiner“, „Kleiner oder gleich“* und Operationen, *„Und“, „Wenn- dann“, „Oder“* und/ oder *„Entweder- oder“* bestehen. Wie z.B. „ *Wenn Herr Amann einen Bart hat, dann hat Herr Bemann eine Glatze. Entweder Herr Amann oder Herr Cemann hat schwarzes Haar.*“
Es ist oft möglich, diese Relationen und Operationen in der Aufgabe fast direkt abzulesen. Solche Logeleien können eine, mehrere oder gar keine Lösungen haben. Sie können auch genau so viele Angaben enthalten, wie nötig sind, damit die Logelei nur eine Lösung hat.

Es gibt auch eine **andere** Art der Textaufgaben, wie z.B. die folgende Logelei : „ *Ist es unmöglich, dass diese Unmöglichkeit unmöglich ist?* “ oder *„Die Einwohner des Dorfes A lügen immer und die Einwohner des Dorfes B sagen immer die Wahrheit. Was soll der Reisende einen Einwohner fragen, damit er bereits nach der ersten Antwort erfährt, in welchem Dorf er sich befindet?* “. Solche Aufgaben beruhen auf einer etwas anderen Weise des logischen Denkens. Hier ist der Leser gefordert, sich den sprachlichen Ausdruck und seine Bedeutung zu überlegen. „ Was ist eine Unmöglichkeit?“, „ Kann Unmöglichkeit unmöglich sein?“, „ Was gibt es für Fragen, die man stellen könnte, um herauszufinden, ob die andere Person lügt oder nicht? “, „ Welche Frage hat nur eine richtige Antwort?“ Solche Fragen kann er sich stellen, um auf die Lösung zu kommen.

Spielkarten, Domino, Zahlenrätsel, Buchstabenrätsel und viele andere **Spiele** und Aufgaben, die uns fordern, logisch zu denken, gehören auch zu Logeleien. Sie können geschrieben sowie auch gezeichnet sein.
Ein Beispiel dafür ist die Aufgabe 22 aus dem Buch „99 Logeleien von Zweistein“: „ *Zwei Freunde haben einen acht Liter fassenden Eimer voll Wein. Sie wollen ihn sich gerecht teilen, doch es steht ihnen nur ein Drei- und ein Fünf- Liter- Krug zur Verfügung. Wie können die beiden sich helfen*?“ Um diese Aufgabe zu lösen, überlegen wir uns zuerst die Eigenschaften der gegebenen Eimer und Krüge. Mit etwas logischem Denken und der Kenntnis der Eigenschaften, können wir dann auf das Resultat kommen, das im Buch graphisch dargestellt ist:



Lösung der Aufgabe 22 im Buch „99 Logeleien von Zweistein“

2. Formalisieren von Logeleien
2.1. Zahlen und Buchstaben

Um Logeleien zu vereinfachen und zu lösen, schreiben wir alles, was gegeben ist, mit Relationen und Operationen, Buchstaben und Zahlen um.
Um das Richtige durch Buchstaben und Zahlen zu ersetzen, betrachten wir zuerst ihre Eigenschaften.

2.1.1. Buchstaben

Es gibt in der Mathematik zwei Arten von Buchstaben: Variablen und Konstanten.
Variablen können mehrere Werte annehmen, so wie z.B. v in der Gleichung *v^2-4=0* → v=*{2;-2}* oder *x* und *y* in der Formel *y= 2x+5*.
Konstanten dagegen bleiben immer festgelegt, sie sind unveränderlich. Sie können nur einen Wert annehmen, so wie z.B. die Zahl e oder die Zahl **π**.

Für die Logeleien gebrauchen wir Variablen, da wir es mit einer Art von Gleichungen oder Aufgaben zu tun haben, die wir lösen müssen. Wir kennen den Wert nicht und müssen durch das Ausprobieren oder andere Methoden herausfinden, welche Werte die Lösungen sind. Wir wissen auch nicht, ob eine Logelei eine oder mehrere Lösungen hat. So sind unsere Buchstaben, wenn eine Logelei mehrere Lösungen hat (z.B. a=1 oder a=2), auf jeden Fall keine Konstanten. Wir müssen also für den allgemeinen Fall Variablen gebrauchen.
So können wir z.B. *einen Hund* als Variable *h* bezeichnen oder *Klara* als Variable k.

Wir vereinbaren, dass wir fortan Variablen immer nur mit kleinen Buchstaben schreiben.

2.1.2. Zahlen

Wir haben also Variablen, die verschiedene Werte annehmen können. Jetzt brauchen wir noch diese Werte, die unverändert bleiben. Wir verwenden hierfür Zahlen.
Die einfachsten Zahlen in der Mathematik sind die **natürlichen** Zahlen. Wir gebrauchen sie, um auch einige gegebene Informationen in den Logeleien mit ihnen zu bezeichnen.
Wir könnten auch nur mit Variablen arbeiten ohne Zahlen zu verwenden, da wir aber uns eher gewöhnt sind, dass eine Variable eine Zahl annimmt, erhalten wir mit Variablen und Zahlen zusammen eine bessere Übersicht.
Z.B. können wir *klug* als *1* oder *Herr Amann* als 1 bezeichnen.

2.1.3. Variable oder Zahl?

In unserem Fall kann eine Variable für einmal nur **einer** Zahl gleich sein, sie kann nicht gleichzeitig mehreren Zahlen gleich sein.
Wenn k=3 ist, kann k nicht mehr gleichzeitig gleich 4 sein.
Eine Zahl kann dagegen **gleichzeitig einer oder mehreren** Variablen gleich sein.
Wenn 4=r ist, schliesst es nicht aus, dass 4=t ist.
Zahlen können untereinander **nicht** einander gleich sein.
4 kann beispielsweise nicht gleich 5 sein oder gleich 2. 4 ist nur gleich 4.
Variablen dagegen **können** einander gleich sein.
r kann beispielsweise gleich t sein: wenn r=4 dann ist auch t gleich 4.
Wenn x=2 und l=2 sind, dann ist x=l.

Wir haben Aufgaben, in denen Gegenstände oder Wesen oder ihre Eigenschaften relativ zueinander stehen. Wenn wir z.B. 4 Subjektive und ihre 4 Eigenschaften haben, können wir 4 Variablen und 4 Zahlen nehmen.
„Die Katzen Aïscha, Binna, Cella und Dora waren auf der Jagd und haben eine Maus, einen Vogel, eine Fliege und eine Ratte nach Hause gebracht. Jede Katze hat je ***ein*** *Opfer mitgebracht.“* Wir haben 4 Katzen und ihre 4 Opfer. Es spielt hier keine Rolle, ob wir Katzen oder Opfer mit Variablen bezeichnen, da ihre Anzahl gleich ist. Wir müssen einfach entweder **alle** Katzen oder **alle** Opfer mit Variablen bezeichnen. Wenn wir Katzen mit Variablen bezeichnen, dann bezeichnen wir **alle** Opfer mit Zahlen. Bzw. umgekehrt, wenn Katzen Zahlen sind, dann sind Opfer Variablen.

Wenn wir zu unserem oben beigebrachten Beispiel zurückkommen: wir haben 4 Männer, 4 ihre Namen und 4 ihre Frauen. Die Hauptpersonen in dieser Aufgabe sind Männer, sie haben je 2 Eigenschaften. Spielt es jetzt eine Rolle, was wir mit Variablen und was mit Zahlen darstellen? Bezeichnen wir beispielsweise die 4 Männer mit Variablen und 8 Frauen- und Männernachnamen mit Zahlen. Jede Variable hat dann 2 Eigenschaften, ist also gleichzeitig zwei verschiedenen Zahlen gleich. Dies ist aber nach unserer Definition unmöglich. Wir müssen dann dies umkehren: Männernamen mit Zahlen bezeichnen und ihre „Eigenschaften“ mit Variablen. Jede Zahl ist dann gleich zwei Variablen, was möglich ist.
Daraus folgt, dass die „Haupt- Gegenstände/-Personen/-Wesen“, die in kleinerer Zahl vorkommen, mit Zahlen zu bezeichnen sind und ihre Eigenschaften mit Variablen.

Graphisch sieht unsere Logelei in der Rohform so aus:

Eigenschaften/ Männer	e(Erich)	f(Fritz)	g(Gustav)	h(Hein-rich)	i(Inge)	j(Josefa)	k(Klara)	l(Luise)
1 (Amann)								
2 (Bemann)								
3 (Cemann)								
4 (Demann)								

„Herr Xerox kommt, Frau Hocker kommt nicht, Herr Brokkoli kommt.“ Wir haben 2 verschiedene Eigenschaften (kommt – kommt nicht) und 3 Personen. So bezeichnen wir diejenigen oder das, was in kleinerer Zahl vorkommt, mit Zahlen. Das sind Eigenschaften. So kann die Eigenschaft „kommt“ zu *1* werden und „kommt nicht“ zu *2*. Die Personen sind in diesem Fall mit Variablen zu bezeichnen.

2.1.4. Aussagen

Eine Aussage in den Logeleien jener Art, mit der wir uns beschäftigen, enthält im Vergleich zu einer Variable immer mindestens eine Relation. Um eine Aussage von einer Variable, die wir früher gebraucht haben, zu unterscheiden, schreiben wir fortan alle Aussagen **immer groß**. In unseren Logeleien enthält die kleinste Aussage eine Relation, eine Variable und eine Zahl oder eine Relation und 2 Variablen. *Beispiele:* $a=2 \Leftrightarrow G$, $b=g \Leftrightarrow K$.
Wenn eine Aussage logische Operationen enthält, kann sie als eine noch grössere Aussage dargestellt werden. *Beispiel:* $(D \text{ und } F) \Leftrightarrow H$
Wir können die Aussage „*Cemann heisst Gustav“* mit einem Buchstaben bezeichnen, beispielweise mit *A*. Ist nun „*Demann ist mit Luise verheiratet“* die Aussage B, dann ist die Aussage „*Entweder ist Demann mit Luise verheiratet, oder Cemann heisst Gustav.*“ mit „*Entweder A oder B*“ zu bezeichnen. Gleichzeitig gilt „*Entweder A oder B*“ als eine neue Aussage, die wir durch C ausdrücken könnten.

2.2. Relationen und Operationen

Um Logeleien formal zu lösen, bezeichnen wir das Gegebene mit Variablen oder Zahlen. Doch um weiter mit ihnen arbeiten zu können, ist es unbedingt nötig, gegebene Informationen mit den richtigen Zeichen miteinander zu verknüpfen.

2.2.1.1. Relation „Gleich“ - „Ungleich“

Die am häufigsten gebrauchte Relation ist „G**leich“**. In der Sprache entspricht sie den Verben *„sein“, „gehören“, „haben“* etc. Wir wissen beispielsweise, dass Herr „5“ einen Staubsauger „s“ hat, dann können wir sagen: „s gehört 5“, also „*s **gleich** 5*“. Oder wir können aber auch sagen „5 hat s“, also „*5 **gleich** s*“, was auch das Gleiche bedeutet. Wenn wir „*Cemann heisst Gustav“* haben, dann können wir sagen, dass der Nachname Gustav dem Cemann gehört, also “g ***gleich** 3*“.

*In der Logik im Programm „Mathematica“ wird die logische Operation „**Gleich**“ als* **„==“** *gebraucht. (z.B. „3 gehört g“ wird in Mathematica so geschrieben: ”3==g”)*

Die Relation „**Ungleich**“ wird dann gebraucht, wenn etwas ausgeschlossen wird. Z.B. Herr „2“ trinkt kein Bier „b“ kann man dies mit *„2 ungleich b*“ bezeichnen. Oder man kann auch sagen, „das Bier „b“ wird vom Herr „2“ nicht getrunken“, d.h. „*b ungleich 2*“.

*Im Programm „Mathematica“ wird die logische Operation „**Ungleich**“ als* **„≠“** *gebraucht. (z.B. „a gehört keine 2“ wird in Mathematica so geschrieben :” a≠2”)*

2.2.1.2 Relationen „Weniger/kleiner“ - „Mehr/grösser“

Manchmal kommt es in den Logeleien vor, dass Frau „a“ glaubt, sie sei jünger als Frau „b“, oder dass Katze „l“ mehr Mäuse gefangen hat, als Katze „u“. Um solche Angaben mit logischen Operationen aufschreiben zu können, gebrauchen wir „**Grösser**“ -„**Kleiner**“ -Zeichen. Diese Zeichen kommen anstelle von „Gleich “/ „Ungleich “- Zeichen.

Im Programm „Mathematica“ wird die logische Operation „Grösser“ als „>“ gebraucht und „Grösser oder gleich“ als „≥“. „Kleiner“ wird als „<“ gebraucht und „Kleiner oder gleich“ als „≤“. (z.B. „Herr x hat weniger als 10000 Haare auf dem Kopf“ wird in Mathematica so geschrieben :” x<10000 ” . „Die Katze l hat mehr Mäuse gefangen als die Katze „u““ wird in Mathematica als ” l>u” geschrieben.)

2.2.2. Logische Operationen „Und“, „Oder“, „Entweder - oder“, “Wenn - dann“

Um eine der weiteren Operationen besser beschreiben zu können, zeichnen wir eine Tabelle.
A und **B** sind zwei Aussagen, denen je ein Wert zugeordnet wird: entweder „**0**“ oder „**1**“. „**0**“ bedeutet, dass die Aussage falsch ist, und „**1“ -** richtig. B- Werte werden vertikal und A – Werte horizontal in der Tabelle dargestellt (blau). Werte für eine entsprechende Operation (analog „1“ oder „0“) kommen an der Stelle der rot markierten Buchstaben.

Operation		**A**	
B	Wert	1	0
	1	A=1 B=1	A=0 B=1
	0	A=1 B=0	A=0 B=0

2.2.2.1 „Und“

Am zweithäufigsten wird in den Logeleien die logische Operation „**Und**“ verwendet. Diese Operation ist aus der Sprache gut bekannt und sie hat auch im Programm *Mathematica* fast dieselbe Bedeutung.
Machen wir ein Beispiel: Aussage A sei „ *Herr Regenschirm spaziert mit seinem Hund*“ und Aussage B „ *Es regnet schon den ganzen Tag*“. Jetzt bilden wir „A **und** B“: Es ergibt sich ein neuer Satz: “*Herr Regenschirm spaziert mit seinem Hund und es regnet schon den ganzen Tag*“. Aussage A kann richtig oder falsch sein. Wenn A falsch ist, dann stimmt der neue Satz nicht mehr: “ *Herr Regenschirm spaziert* ***nicht*** *mit seinem Hund und es regnet schon den ganzen Tag*“ , wenn aber A richtig ist, dann kommt es auf die Aussage B an. Falls Aussage B falsch ist, stimmt der neue Satz trotzdem nicht: “*Herr Regenschirm spaziert mit seinem Hund und es regnet* ***nicht*** *den ganzen Tag*“, wenn jedoch B richtig ist, stimmt der neue Satz. Also, der neue Satz stimmt **dann und nur dann**, wenn A **und** B richtig sind.
Anhand dieses Beispiels füllen wir die Tabelle aus. Die roten Werte „0“ und „1“ entsprechen dem Wert der ganzen Aussage „A **und** B“ (in unserem Beispiel “ *Herr Regenschirm spaziert mit seinem Hund* ***und*** *es regnet schon den ganzen Tag*“)
Wenn z.B. A =„1“ (richtig) und B =„0“ (falsch) sind, dann ist die Aussage „A und B“ = „0“, also falsch.

Operation „A **und** B“	**A**		
B	Wert	1	0
	1	1	0
	0	0	0

*Im Programm „Mathematica“ wird die logische Operation „**Und**“ als Zeichen „**&&**“ gebraucht. (z.B. (A und B) wird in Mathematica so geschrieben: ”(A**&&**B)”*
(a=1 und b=2) folgendermassen:”(a==1&&b==2)”)

2.2.2.2. „Oder“

Es gibt zwei verschiedene „**Oder** “- Operationen : einfach „Oder“ und „Entweder- oder“. Um Logeleien zu lösen ist es sehr wichtig, zwischen den beiden unterscheiden zu können.
Zuerst schauen wir uns die „Oder“ - Operation genauer an.
Nehmen wir an, Aussage A sei „ *Eskimoasen haben kalt* “, und B „*Sudaner haben heiss*“. Nun machen wir eine neue Aussage „A **oder** B“: „*Eskimoasen haben kalt oder Sudaner haben heiss*“. Wenn A richtig ist, dann stimmt die neue Aussage bei allen B: „*Eskimoasen haben kalt*“, egal ob *Sudaner heiss haben oder* ***nicht***. Wenn jedoch A falsch ist („ *Eskimoasen haben* ***nicht*** *kalt* “), dann ist es von B abhängig, ob „A oder B“ stimmt oder nicht. Wenn „*Sudaner* ***nicht*** *heiss haben*“ und „*Eskimoasen* ***nicht*** *kalt haben*“, **dann und nur dann** stimmt die neue Aussage **nicht**. Damit also „A oder B“ stimmt, reicht es bereits aus, dass eine dieser beiden Aussagen richtig ist. Treffen beide Aussagen zu, so ist der neue Satz auch richtig.
Wir zeichnen wieder eine Tabelle wie bei „Und“, und ersetzen die roten Werte, durch die neuen Werte, die der Operation „Oder“ entsprechen.

Operation „A **oder** B“	**A**		
B	Wert	1	0
	1	1	1
	0	1	0

Im Programm „Mathematica" wird die logische Operation „Oder" als Zeichen **„||"** *gebraucht. (z.B. (A oder B) wird in Mathematica so geschrieben :"(A||B)";*
(a=1 oder b=2) folgendermassen :"(a==1||b==2)")

2.2.2.3. „Entweder - Oder"

In unserer Logelei hatten wir *„Entweder ist Demann mit Luise verheiratet, oder Cemann heisst Gustav.". „Demann ist mit Luise verheiratet"* sei Aussage A, *„Cemann heisst Gustav."* sei B. Wir haben somit die ganze Aussage „**Entweder A oder B**". Wenn *„Demann mit Luise verheiratet ist"*, also A, richtig ist, **dann und nur dann** stimmt die ganze Aussage, wenn *„Cemann **nicht** Gustav heisst."*, wenn also B falsch ist. Wenn *„Demann **nicht** mit Luise verheiratet ist"*, dann muss *„Cemann Gustav"* heissen. Die Aussage „Entweder A oder B" ist somit bei A=0 **dann und nur dann** richtig, wenn B=1 ist.

Operation „**Entweder** A **oder** B"	**A**		
B	Wert	1	0
	1	0	1
	0	1	0

Im Programm „Mathematica" wird die logische Operation „Entweder- Oder" als **Xor[..,..]** *gebraucht. (z.B. (Entweder A oder B) wird in Mathematica so geschrieben :"***Xor[***A,B***]***";*
(entweder a=1 oder b=2 oder c=3) folgendermassen : "Xor[a==1,b==2,c==3]")

2.2.2.4 „Wenn - Dann"

Nehmen wir an, die Aussage A sei „ *Fritz hat seinen 5-Liber verloren* " und B „ *Fritz kauft jetzt kein Sneakers*". Machen wir eine neue Aussage „**Wenn** A, **dann** B": „*Wenn Fritz seinen 5-Liber verloren hat, dann kauft er jetzt kein Sneakers*".
Wenn A richtig ist („ *Fritz hat seinen 5-Liber verloren* "), stimmt der neue Satz **dann und nur dann** wenn B auch richtig ist („ *Fritz kauft jetzt kein Sneakers*"). Denn Fritz kann kein Sneakers kaufen, wenn er sein Geld verloren hat.
Wenn A falsch ist („ *Fritz hat seinen 5-Liber **nicht** verloren* "), dann kann Fritz ein Sneakers kaufen oder auch nicht. In beiden Fällen würde es nicht gegen die neue Aussage verstossen. Daraus folgt, dass diese Aussage nicht gleich „0" ist, also „1".
(Achtung: „Wenn Fritz seinen 5-Liber verloren hat, dann kauft er jetzt kein Sneakers" ist nicht gleich dem Satz : „Wenn Fritz jetzt kein Sneakers kauft, dann hat er seinen 5-Liber verloren")

Operation „**Wenn** A , **dann** B"	**A**		
B	Wert	1	0
	1	1	1
	0	0	1

Im Programm „Mathematica" wird die logische Operation „Wenn-dann" als **Implies[..,..]** *gebraucht. (z.B.(Wenn A Dann B) wird in Mathematica so geschrieben: "***Implies[***A,B***]***";*
(wenn a=1 dann b=2) wird in Mathematica so geschrieben: "Implies[a==1,b==2]")

2.3. Relationen- und Operationen-Gegenteile

Als **Relations-** und **Operationsgegenteil** bezeichne ich eine neue Relation bzw. eine neue Operation, bei der genau all diejenigen Aussagen richtig sind, die bei der Ursprungsrelation bzw. Ursprungsoperation falsch sind. Diese ursprüngliche und neue Relation bzw. Operation sind Gegenteile voneinander.

2.3.1.1 „Gleich“ - „Ungleich“ - Gegenteile

Jede der zwei Operationen „Gleich“ und „Ungleich“ ist genau das Gegenteil der anderen nach der Definition. Wie wir auch sehen können, besteht „ungleich“ aus „**un**-“, was ein Gegenteil bedeutet, und „-gleich“.
Wir haben beispielsweise die Ursprungsoperation ($x+b=a$), daraus folgt ($x = a-b$). Es ist vorausgesetzt, dass x von minus unendlich bis plus unendlich sein kann. Die einfachste und sicherste Methode, eine Gleichung zu finden, die genau alle Lösungen ausser($x=a-b$) enthält, ist, in der ersten Gleichung anstelle von „gleich“ „ungleich“ zu setzen. Dann ist das Gegenteil von dieser Operation ($x+b\neq a$). Diese Gleichung hat die Lösung $x \in[-\infty; a-b)\cup(a-b; +\infty]$.

2.3.1.2 „Weniger/Kleiner“ - „Mehr/Grösser“ - Gegenteile

Diese zwei Operationen sind auch *fast* Gegenteile voneinander. Dabei sollen wir beachten, dass wenn x beispielsweise grösser oder kleiner als y ist, dann ist x **nicht gleich** y.
*Z.B. $a>b$ (a- eine ganze Zahl) bedeutet, dass ($a=b+1, b+2, b+3, b+4, b+5, .., +\infty$) sein kann, also **nicht gleich oder kleiner** als b. Die Operationen „>“ und „≤“ sind somit Gegenteile.*
*Wenn $a<b$ ist, bedeutet dies, dass ($a= -\infty, ..., b-5, b-2, b-1$) sein kann, also **nicht gleich oder grösser** als b. Die Operationen „<“ und „≥“ sind somit auch Gegenteile voneinander.*

2.3.2.1. „Und“ –Operationsgegenteil

Viele logische Operationen, die uns im Alltag begegnen, nehmen wir auf der unbewussten Ebene wahr und können sie nur mit Mühe genau beschreiben. Darum versuchen wir es jetzt mit Hilfe der Tabelle zu tun, die wir zuvor ausgefüllt haben.
Um Operationengegenteil von „A und B“ zu finden, ersetzen wir in der Tabelle „A und B“ alle roten „1“ durch „0“ und alle roten „0“ durch „1“.

Operationsgegenteil von „A **und** B“	**A**		
B	Wert	1	0
	1	0	1
	0	1	1

Wenn wir uns jetzt diese Tabelle genauer anschauen, bemerken wir, dass Lösungen genau bei **allen** *A=0* **oder** *B=0* auftreten. Es ist eine Art Vereinigung von Lösungen für *alle A=0 und alle B=0*, nur eine Lösung ist ausgeschlossen: bei (*A=1 und B=1)*.
Jetzt vergleichen wir diese Tabelle mit denjenigen, die wir schon vorher erstellt haben. Bei der „Oder“ –Operation ist es auch so, dass 3 Lösungen richtig sind und eine falsch ist. Der Unterschied besteht darin, dass bei der „Oder“ - Operation die falsche Lösung bei (*A=0 und*

B=0) ist, und hier haben wir die falsche Lösung bei (*A=1 und B=1)*. Dies entspricht dann der Operation (A=0 *oder* B=0).
Das Gegenteil von (A=*1 und B=1*) ist somit (*A=0 oder B=0*), also (*A≠1 oder B≠1*).
Wir können dies so aufschreiben:
„*A==1&&B==1*" ist das Gegenteil von „*A≠1||B≠1*", oder im Allgemeinen :

(A && B) ist das Gegenteil von ((A-Gegenteil) || (B-Gegenteil))

*Z.B. (a==3**&&**b==2) ist das Gegenteil von (a≠3||b≠2);*
*(a==3**&&**b≠5) ist das Gegenteil von (a≠3||b==5).*

2.3.2.2. „Oder" -Operationsgegenteil

Ich habe im Kapitel 2.3.1.2. ein Beispiel gegeben, in dem „Oder" - Gegenteil vorkam: „*a>b, bedeutet, dass a **nicht gleich oder kleiner** als b sein kann.". „a **nicht (gleich oder kleiner)** als b sein kann*" bedeutet, dass *a* **nicht gleich** *b* **und nicht kleiner** als *b* sein kann. Um dieses Resultat mit den mathematischen Auswertungen zu vergleichen, erstellen wir wieder eine Tabelle, analog der „Und" - Tabelle. Wir haben in der „A Oder B" – Operation - Tabelle alle roten „1" durch „0" und alle roten „0" durch „1" ersetzt.
Wir sehen, dass diese Operation nur eine Lösung hat: bei (*A=0* **und** *B=0)*, oder anders geschrieben (A *≠1&&B ≠1*). Unser Resultat stimmt mit der oben sprachlich formulierten Lösung überein.

Operationsgegenteil von „A **oder** B"	**A**		
B	Wert	1	0
	1	0	0
	0	0	1

(A || B) ist das Gegenteil von ((A-Gegenteil) && (B-Gegenteil))

*Z.B. (a==6||d≠2) ist das Gegenteil von (a≠6**&&**d==2);*
*(a>2||b==3) ist das Gegenteil von (a≤2**&&**b≠3).*

2.3.2.3 „Entweder - Oder" -Operationsgegenteil

Eine Operation, die lauten würde „Nicht (Entweder A Oder B)" kann man sich schwer vorstellen. Darum stellen wir sie nur abstrakt dar.
Wir sehen, dass das Gegenteil der Operation „Entweder- Oder" zwei „1" und zwei „0" hat, so wie die Operation „Entweder- Oder".
Der Unterschied besteht darin, dass hier bei gleichen A und B eine „1" ist. Sind jene ungleich, ergibt dies eine „0".

Operationsgegenteil von „**Entweder** A-**Oder** B"	**A**		
B	Wert	1	0
	1	1	0
	0	0	1

Wir können es auf verschiedene Arten aufschreiben:
(A==1&&B==1)||(A==0&&B==0) ⇔
(A==1&&B==1)||(A≠1&&B≠1) ⇔
Xor[A==1&&B==1,A≠1&&B≠1]

Das Gegenteil von der "Entweder- Oder" –Operation Xor [A, B] ist Xor [A &&B, A-Gegenteil && B-Gegenteil]

Z.B. Xor [a==6,d≠2] ist das Gegenteil von Xor [a==6&&d≠2,a≠6&&d==2]; Xor [a>2,b==3] ist das Gegenteil von Xor[a>2&&b==3, a≤2&&b≠3].

2.3.2.4. „Wenn - Dann" -Operationsgegenteil

Das Operationsgegenteil von „Wenn - Dann" wird praktisch nie verwendet.
In der Tabelle sieht man gut, dass es nur eine Lösung hat: bei *(A=1 und B=0).*
Somit können wir **das Gegenteil der „Wenn - Dann" –Operation Implies[A, B]** durch **(A==*1&&B≠1)* oder auch**
((A) && (B-Gegenteil))
definieren.

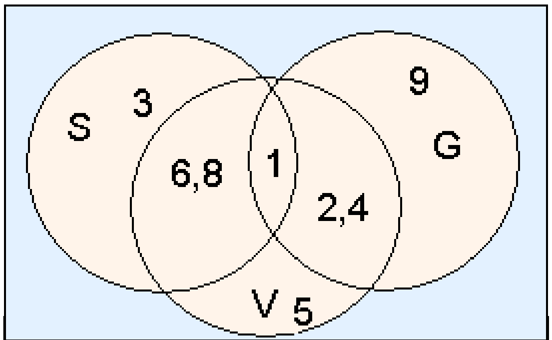
Operationsgegenteil von „**Wenn** A-**Dann** B"	**A**		
B	Wert	1	0
	1	0	0
	0	1	0

Z.B. Implies [a==6,d≠2] ist das Gegenteil von (a==6&&d==2); Implies [a>2,b==3] ist das Gegenteil von (a>2&&b≠3)

2.4. Umformung von „Wenn- Dann“ - und „Entweder- Oder“ - Operationen und weitere Regeln

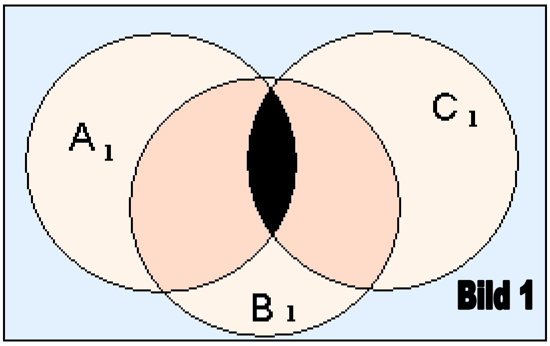
Die Operationen „Wenn- Dann“ und „Entweder- Oder“ lassen sich mit Hilfe der anderen logischen Operationen umschreiben.
Es ist möglich, alle Operationen in den Logeleien zum Schluss nur mit *„==“, „||“* und *„**&&**“* auszudrücken. Da es aber oft mit sehr grossem Aufwand verbunden ist, lohnt es sich nicht immer. Darum gebrauchen wir meistens auch noch *„≠“, „>“* und *„<“*.

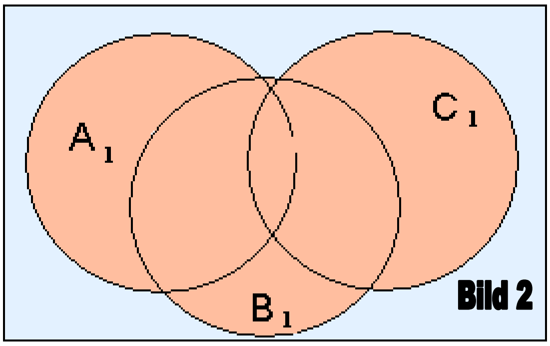
***Grundlagen**: Wenn es zwischen mehreren Aussagen „Und“ steht, bedeutet dies, dass **alle** diese Aussagen **zusammen** eine grössere Aussage darstellen. Die Lösung so einer grossen ganzen Aussage ist diejenige, die in **jedem** Mitglied der Aussage der Art „A && B“ vorkommt. Z.B. für die Aussage A**&&**B**&&**C gibt es nur eine Lösung: bei A -richtig, B -richtig und C-richtig.*
*Wenn es zwischen den Aussagen „Oder“ steht, bedeutet dies, dass **jedes** „Mitglied“ allein schon eine Lösung sein **kann,** und dass **ein Mitglied** auf jeden Fall die Lösung **ist**. Z.B. die Aussage A||B||C hat alle Lösungen mit A –richtig, B –richtig und C –richtig, die Lösungen von einzelnen „Mitgliedern“ werden sozusagen zusammengezählt.*
Jetzt stellen wir es graphisch dar. A1,B1 und C1 seien Mengen der Lösungen von der Aussagen A, B und C. Diese Mengen schneiden sich dort, wo sie die gleichen Lösungen haben.
***Anwendbeispiel**: Menge S hat Lösungen {1,3,6,8}, V hat Lösungen{1,2,4,5,6,8} und G {1,2,4, 9}.Aussage“ S **und** V“ hat 3 gemeinsame Lösungen: {1},{6}und {8}. Diese Lösungen befinden sich in dem Teil, wo sich diese zwei Mengen schneiden. Aussage „S **oder** V“ hat 7 Lösungen: {1,2,3,4,5,6,8}: alle Lösungen von S und alle Lösungen von V. „S **und** V **und** G“ hat nur eine gemeinsame Lösung: {1}. „S **oder** V **oder** G“ hat dagegen alle 8 Lösungen: {1,2,3,4,5,6,8,9}.*



Anwendbeispiel Abbildung

*Auf dem linken Bild 1 sehen wir die graphische Darstellung der Lösungsmengen der Aussage „A1 **und** B1 **und** C1“. Der **schwarz** gefärbte Teil ist der Teil, wo sich alle Lösungsmengen schneiden. Somit ist dieser Teil auch die Lösung dieser ganzen grossen Aussage. Auf dem rechten Bild 2 ist die Lösungsmenge der Aussage „A1 **oder** B1 **oder** C1“ zu sehen. Alles, was beige angefärbt ist, ist die Lösung.*





2.4.1.1. „Implies“ („Wenn – dann“)

Wir schauen uns zuerst alle Werte in der Tabelle an, die möglich sind. Wir stellen fest, dass bei ***allen*** *B=1* **und** bei ***allen*** *A=0* die Operation *Implies* richtig ist. *(Manchmal verwirrt die Sprache: wenn es „Und“ steht, bedeutet dies nicht zwangsläufig, dass da einfach „&&“ gebraucht werden kann.)* Wenn es von allen *B=1* und allen A= 0 die Rede ist, dann haben wir es hier mit einer Art Vereinigung der Lösungen zu tun:

Operation „**Wenn** A **dann** B“	A		
B	Wert	1	0
	1	1	1
	0	0	1

(A==1 und B==1) oder (A==0 und B==1) oder (A==0 und B==0) ⇔ *(A==1&&B==1)||(A==0&&B==1)||(A==0&&B==0)* ⇔ *(B==1)||(A==0)* *(B==1 umfasst 2 Lösungen: A=0 und A=1, A=0 umfasst noch einmal eine Lösung von B=1 und noch eine Lösung von B=0)*

Implies[A, B] ⇔ ((A-Gegenteil) || B)

Z.B. Implies[a==1, b==3] ⇔ (a≠1||b==3) (⇔ Implies[b≠3,a≠1]) ;
Implies[a>2, b≠2] ⇔ (a≤2||b≠2) (⇔ Implies[b==2,a≤2]).
In unserem Beispiel hatten wir:„Wenn Luises Mann Fritz heisst, dann ist der Vorname von Klaras Mann nicht Gustav.“ Diese Aussage ist also gleichwertig wie die Aussage „Luises Mann heisst nicht Fritz oder der Vorname von Klaras Mann ist nicht Gustav.“

2.4.1.2. „Xor“ („Entweder - Oder“)

Über diese Operation wissen wir, dass sie bei *(A=0 und B=1)* oder bei *(A=1 und B=0)* richtig ist. Wir können schreiben: *(A≠1&&B==1)||(A==1&&B≠1)*, oder auch : *Implies[A==0,B==1]***&&***Implies[A==1,B==0]*⇔ *Implies[A≠1,B==1]***&&***Implies[A==1,B≠1]*

(Im ersten Beispiel verwenden wir „||“, weil (A≠1&&B==1) schon eine fertige Lösung ist, sowie auch (A==1&&B≠1). Diesen 2 Lösungen sind ***unabhängig*** *voneinander. Zwischen Implies verwenden wir „&&“, weil alle Implies zusammen einander* ***ergänzen*** *und eine neue grössere Aussage bilden).*
Implies[A≠1,B==1]&&Implies[A==1, B≠1] ⇔ (A==1||B==1)&&(A≠1||B≠1)

Operation „**Entweder** A **Oder** B“	A		
B	Wert	1	0
	1	0	1
	0	1	0

Xor[A, B] ⇔ ((A || B) && (A-Gegenteil || B-Gegenteil)) ⇔ ((A && B-Gegenteil) ||(B && A-Gegenteil))

Z.B. Xor[a==3, c==4] → (a==3||c==4)&&(a≠3||c≠4) (→Xor[a≠3,c≠4]) ;
Xor[a≠5,g≠3] → (a≠5||g≠3)&&(a==5||g==3) (→Xor[a==5,g==3]).
In unserem Beispiel hatten wir: „Entweder ist Amanns Vorname Heinrich, oder Bemanns Frau heisst Inge.“ Dann ist diese Aussage gleichwertig wie die Aussage „(Amanns Vorname ist Heinrich und Bemanns Frau heisst nicht Inge) oder (Amanns Vorname ist nicht Hinrich und Bemanns Frau heisst Inge).“

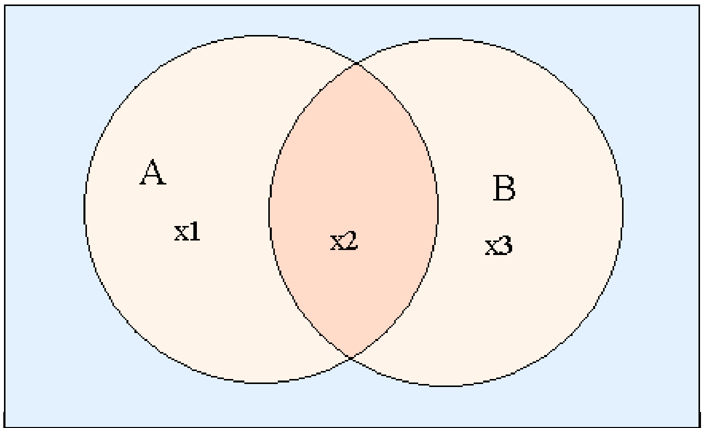
Betrachten wir die Situation, wenn es mehrere Aussagen in der Operation „Entweder- Oder“ vorhanden sind. Wir haben beispielsweise folgende Bedingung: „Entweder A Oder B Oder C“. Hier können wir unsere Tabelle in ihrer ursprünglichen Form nicht gebrauchen. Deshalb zeichnen wir eine neue Tabelle, die drei Aussagen enthält und uns zeigt, bei welchen Werten die Operation stimmt.
Die Operation „Entweder A oder B oder C“ stimmt bei den folgenden Aussagen: A-richtig **und** (B **und** C) - falsch, B-richtig **und** (A **und** C) - falsch, C-richtig **und** (A **und** B) - falsch. Wir können es folgendermassen aufschreiben:

Operation „**Entweder** A **Oder** B **Oder** C“	**A**	**B**	**C**
0	1	1	1
0	1	1	0
1	1	0	0
0	1	0	1
0	0	0	0
1	0	0	1
1	0	1	0
0	0	1	1

Xor[A,B,C] ⇔
((A &&B-Gegenteil &&C-Gegenteil)||
(B &&A-Gegenteil &&C- Gegenteil)||
(C &&A-Gegenteil &&C- Gegenteil)) ⇔
((A||B||C) &&
(A-Gegenteil || B-Gegenteil)&&
(B-Gegenteil || C-Gegenteil)&&
(A-Gegenteil || C-Gegenteil))

2.4.2.1. „Und “ zwischen mehreren Aussagen

Jetzt betrachten wir die Fälle, in denen mehrere Aussagen durch „Und“ verbunden sind.
Zuerst nehmen wir die einfachste Aussage A&&B. Wir stellen sie graphisch dar. {x1, x2} sind Lösungsmengen von A und {x2, x3} sind Lösungsmengen von B. Wenn wir Lösungsmengen von A&&B und B&&A betrachten, merken wir, dass die beiden gleich sind und nur eine Lösungsmenge x2 haben. Es spielt also keine Rolle, in welcher Reihenfolge diese zwei Aussagen vorkommen, sie haben immer das Gleiche als Resultat.

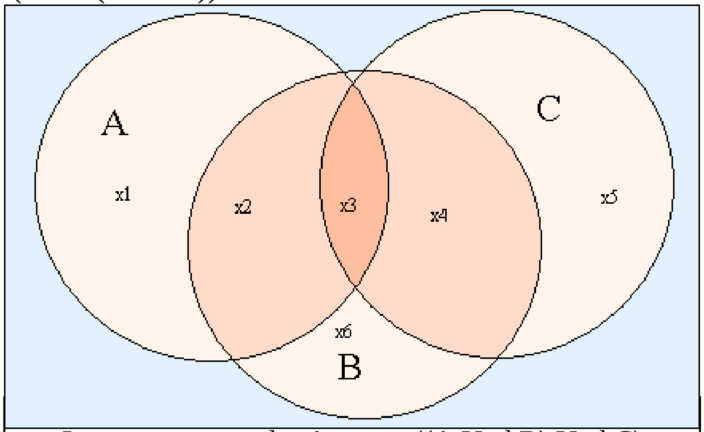


Lösungsmengen der Aussage (A Und B)

Als Nächstes betrachten wir Situationen, in denen auch noch Klammern vorkommen. Das einfachste Beispiel ist (A&&B)&&C. Wir stellen es analogisch graphisch dar (S. 17). Wenn wir Lösungsmengen von ((A&&B)&&C), ((B&&A)&&C), ((A&&C)&&B), ((B&&C)&&A), ((C&&A)&&B) und ((C&&B)&&A) vergleichen, sehen wir, dass sie alle gleich sind (x3). Es spielt also keine Rolle, in welcher Reihenfolge wir die Aussagen nehmen, wir kommen **immer** auf das gleiche Resultat, und zwar auf die Lösungsmenge, wo sich **alle** Lösungsmengen **schneiden**.

Jede der Aussagen A, B und C kann als eine komplexere Aussage dargestellt werden. B ist beispielsweise (D&&E), dann ergibt dies (A&&(D&&E))&&C. Wir schreiben zuerst (A&&(D&&E)) um:

((A&&D)&&E). Beim zweiten Schritt bezeichnen wir ((A&&D)&&E) als eine neue Aussage H und fügen „&&C" dazu: (H&&C). Wir können schreiben: (A&&(D&&E)&&C) ⇔ ((A&&D)&&E)&&C. Für jede der Aussagen können wir eine komplexere nehmen (mit „Und"), jedoch das Prinzip bleibt gleich. Denn **alle komplexere** Aussagen dieser Art können in die Form (A&&B)&&C oder A&&B gebracht werden (nach Definition) und dann mit Hilfe unserer Regeln **schrittweise** ausgeklammert werden.



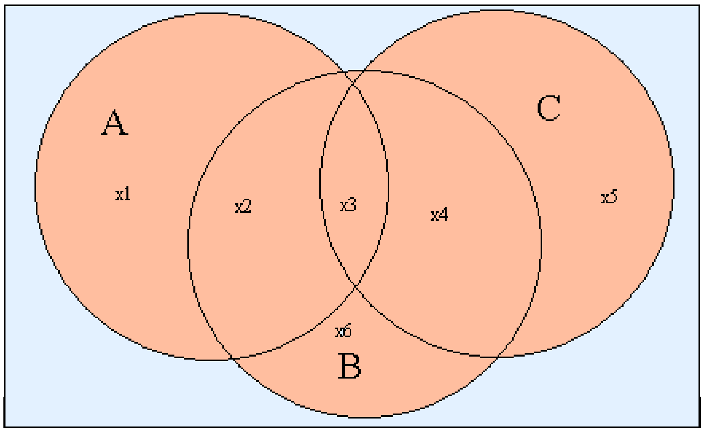
Lösungsmengen der Aussage ((A Und B) Und C)

Die Reihenfolge der Aussagen zwischen „Und" spielt keine Rolle: Man kann sie beliebig verändern, bzw. auf die Klammern verzichten.

Z.B. (A&&B)&&V&&(G&&H) ⇔ A&&B&&V&&G&&H ⇔ A&&(B&&V)&&G&&H; a==3&&b==4&&h==5&&g≠2 ⇔ a==3&&(b==4&&h==5)&&g≠2

2.4.2.2. „Oder " zwischen mehreren Aussagen

Können wir auch die Reihenfolge der Aussagen zwischen mehreren „Oder" verändern? Um diese Frage zu beantworten greifen wir zu einem analogischen Graph wie bei „Und". Wir nehmen 3 Aussagen *(A, B, C)* und verbinden sie durch „Oder": *(A||B||C).* (A||B) hat als Lösungen: {x1, x2, x3, x4, x6}. ((A||B)||C) hat alle vorherigen Lösungen plus diejenigen welche in der Menge C vorkommen: {x3, x4, x5}. Wir zählen also alle Lösungsmengen zusammen. Auch wenn wir die Reihenfolge der Aussagen A, B und C ändern , kommen wir immer auf das gleiche Resultat. Somit gelten für die drei Aussagen zwischen zwei „Oder" genau die gleichen Regeln wie für „Und". Um zu beweisen, dass E||D gleichwertig wie die Aussage D||E ist, betrachten wir das Folgende: (A||B)||C ist gleichwertig wie C||(A||B). Jetzt sei C die Aussage D und (A||B) – E. Wir haben: E||D ⇔ D||E, was wir auch beweisen wollten. Das Prinzip bei der Operation „Oder" ist somit genau gleich wie bei der Operation „Und":



Lösungsmengen der Aussage ((A Oder B) Oder C)

Die Reihenfolge der Aussagen zwischen „Oder" spielt keine Rolle: Man kann sie beliebig verändern, bzw. auf die Klammern verzichten.

Z.B.(A||B||V||G||H) ⇔ (A||B||V ||H||G) ⇔ (A||(B||V)||H||G); (a==3||(b==4||h==5)||g≠2) ⇔ (a==3||b==4||h==5||g≠2)

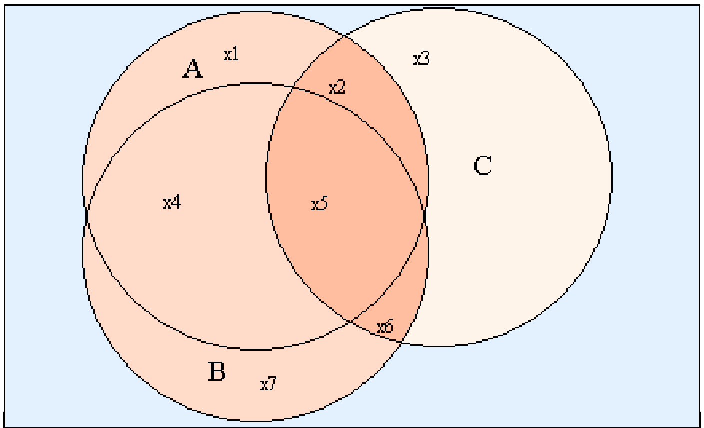
2.4.2.3. „Und “ zwischen mehreren Aussagen der Art „A oder B“

In der Logelei wird fast immer die Operation „Und“ zwischen mehreren ganzen Aussagen verwendet. Denn diese Aussagen **ergänzen** im Normalfall einander und bilden alle **zusammen eine** grosse Aufgabe. Jede ganze getrennte Aussage wird in Klammern gesetzt.

*Für unsere am Anfang beigebrachte Logelei gilt: „(Entweder ist Amanns Vorname Heinrich, oder Bemanns Frau heisst Inge)**, und**,(wenn Cemann mit Josefa verheiratet ist, dann – und nur in diesem Falle – heisst Klaras Mann nicht Fritz), **und**, (wenn Josefas Mann nicht Erich heisst, dann ist Inge mit Fritz verheiratet), **und** .... “*

Es kommt vor, dass eine oder mehrere Aussagen „Entweder- Oder“ -, „Oder“ - oder „Wenn-Dann“ –Operationen enthalten. Wie wir es bereits gesehen haben, können wir jede dieser Operationen vereinfachen, und zwar bis auf die Form (A||B). Um das Verhalten der eingeklammerten, „Oder“ –Operationen enthaltenden Aussagen zwischen „Und“ zu untersuchen, betrachten wir die einfachste Form (A||B)&&C. Wir stellen sie graphisch dar.

(A||B) hat die Lösungsmengen {x1, x2, x4, x5, x6, x7}. (A||B)&&C hat die Lösungsmengen {x2, x5, x6}. Aus dem Graph sehen wir, dass {x2, x5} der Lösungsmengen der Aussage (A&&C) entsprechen, und {x5, x6} der Aussage (B&&C). Diese Lösungsmengen werden sozusagen zusammengezählt, wir können also „Oder“ dazwischen schreiben: (A&&C)||(B&&C).



Lösungsmengen der Aussage ((A Oder B) Und C)

(A||B)&&C ⇔ (A&&C)||(B&&C)

Mit Hilfe dieser Umformung können wir schrittweise alle komplexere Aussagen zusammen in die Form (H&&L&&...)||(K&&L&&...)||(M&&F&&..)||(...) bringen.

Wir machen einige Beispiele:

(G||H)&&(B&&C) ⇔ *(G||H)&&M, M=B&&C* ⇔ *(G&&M)||(H&&M)* ⇔ ***(G&&(B&&C))||(H&&(B&&C))*** ⇔ *(G&&B&&C)||(H&&B&&C)*

(G||H)&&(L||M) ⇔ *(G||H)&&(X), X=(L||M)* ⇔ *(G&&X)||(H&&X)* ⇔ *(G&&(L||M))||(H&&(L||M))* ⇔ *((G&&L)||(G&&M))||((H&&L)||(H&&M))* ⇔ ***(G&&L)||(G&&M)||(H&&L)||(H&&M)***

(G||H)&&(L||M||K) ⇔ *(G||H)&&(X), X=(L||M||K)* ⇔ *(G&&X)||(H&&X)* ⇔ *(G&&(P||K))||(H&&(P||K)), P=(L||M)* ⇔ *((G&&P)||(G&&K))||((H&&P)||(H&&K))* ⇔ *(G&&(L||M))||(G&&K))||((H&&(L||M))||(H&&K))* ⇔ ***(G&&L)||(G&&M)||(G&&K)||(H&&L)||(H&&M))||(H&&K)***

Wie es in den Beispielen zu sehen ist, können wir die logische Operation „||“ in der Mathematik mit der Addition „+“ vergleichen, und die logische Operation „&&“ mit der Multiplikation „*“.

Diese Anordnung entspricht auch den Regeln der Kombinatorik. Wenn wir „Und“ haben, multiplizieren wir, und wenn wir „Oder“ haben, d.h. alternativen Lösungen, addieren wir.
Beispiel: (G||H)&&(L||M||K), vergleichbar mit (G+H)(L+M+K) ⇔*

*(G*L)+(G*M)+(G*K)+(H*L)+(H*M)+(H*K)* ⇔
(G&&L)||(G&&M)||(G&&K)||(H&&L)||(H&&M))||(H&&K)

Wir wenden diese Regeln für den Ausschnitt aus unserer Logelei 94 an: *„(Entweder ist Amanns Vorname Heinrich, oder Bemanns Frau heisst Inge* ***), und****,(Wenn Luises Mann Fritz heisst, dann ist der Vorname von Klaras Mann nicht Gustav)."* ⇔
„((Amanns Vorname ist Heinrich und Bemanns Frau heisst nicht Inge) **oder** *(Amanns Vorname ist nicht Heinrich und Bemanns Frau heisst Inge)* ***), und****, ((Luises Mann heisst nicht Fritz)* ***oder*** *(der Vorname von Klaras Mann ist nicht Gustav))."* ⇔
„(Amanns Vorname ist Heinrich und Bemanns Frau heisst nicht Inge ***und*** *Luises Mann heisst nicht Fritz)* ***oder*** *(Amanns Vorname ist Heinrich und Bemanns Frau heisst nicht Inge* ***und*** *der Vorname von Klaras Mann ist nicht Gustav)* ***oder*** *(Amanns Vorname ist nicht Heinrich und Bemanns Frau heisst Inge* ***und*** *Luises Mann heisst nicht Fritz)* ***oder*** *(Amanns Vorname ist nicht Heinrich und Bemanns Frau heisst Inge* ***und*** *der Vorname von Klaras Mann ist nicht Gustav)."*

3. Lösen von Logeleien

Jetzt gehen wir zum Kapitel über, wo es darum geht, Logeleien mit Hilfe unserer Theorie zu vereinfachen und zu lösen.
Ich habe 4 Lösungsmethoden selbst entworfen und aufgebaut, die für verschiedene Logeleien dienen sollen. Um alle diesen Verfahren zu präsentieren, habe ich mich für die erste einfache Aufgabe aus dem Buch „99 Logeleien von Zweistein“ entschieden.

3.1. Aufgabe 1 im Buch „99 Logeleien von Zweistein“

„Beim Schulausflug schwatzten die Mädchen viel. Es war von Aki, Bauzi, Knirps und Dicki die Rede. Die Lehrerin, die sich die Unterhaltung eine Weile lang angehört hatte, fragte schliesslich: „Wovon redet ihr eigentlich? “ Eine Schülerin antwortete: „ Von einem Mädchen, einem Jungen, einem Hund und einer Katze. “ Darauf die Lehrerin: „ Und wer ist was? “ Die Mädchen hatten keinen Grund es zu verstecken, aber allzu leicht wollten sie es ihrer Lehrerin auch nicht machen, deshalb gaben sie zur Antwort:
„ Wenn Aki nicht der Junge ist und Bauzi nicht das Mädchen, dann ist Knirps der Hund. “
„ Wenn Dicki nicht die Katze ist, dann ist, falls Aki nicht das Mädchen ist, Bauzi der Hund.“
„ Mindestens eine der folgenden drei Angaben ist richtig: Knirps ist die Katze, Dicki ist der Junge, Aki ist der Hund.“
„ Wenn weder Knirps noch Dicki das Mädchen ist, dann ist Bauzi der Hund. “
„ Und wenn.. “
„ Genug, diese Angaben reichen mir schon “, unterbrach die Lehrerin.
Wie heissen Junge, Mädchen, Hund und Katze.“

Zuerst analysieren wir die Aufgabe. Wir haben 4 Namen und 4 Wesen. Es spielt also keine Rolle, ob wir Namen oder Wesen mit Variablen bezeichnen. Dann:
Aki sei a, Bauzi sei b, Knirps sei c und Dicki sei d;
Das Mädchen 1, Der Junge 2, der Hund 3 und die Katze 4.
Die Namen werden den Wesen zugeordnet, d.h. wir können „gleich“ gebrauchen.

„ Wenn Aki nicht der Junge ist und Bauzi nicht das Mädchen, dann ist Knirps der Hund. “
Daraus folgt, dass wenn a ungleich 2 und b ungleich 1 ist, ist c gleich 3, oder in Mathematica so geschrieben ⇒ ***Implies[a≠2&&b≠1,c==3]***

„ Wenn Dicki nicht die Katze ist, dann ist, falls Aki nicht das Mädchen ist, Bauzi der Hund.“
Wenn A und B, dann C ⇒ ***Implies[d≠4&&a≠1,b==3]***

„ Mindestens eine der folgenden drei Angaben ist richtig: Knirps ist die Katze, Dicki ist der Junge, Aki ist der Hund.“ Eine Angabe stimmt auf jeden Fall, aber es können auch alle stimmen ⇒ ***c==4||d==2||a==3***

„ Wenn weder Knirps noch Dicki das Mädchen ist, dann ist Bauzi der Hund. “ Wenn Knirps nicht das Mädchen ist und Dicki nicht das Mädchen ist, dann ist Bauzi der Hund: ⇒ ***Implies[c≠1&&d≠1,b==3]***

Somit haben wir 4 Aussagen, die wir durch „Und“ miteinander verknüpfen können.

3.1.1. Lösen in Mathematica

Um meine Lösungen mit den Resultaten vergleichen zu können, habe ich das PC-Programm *Mathematica* gebraucht. Dieses Programm kennt alle logischen Operationen und darum ist es bequem zum Anwenden. Es hat aber auch seine eigene Sprache, die man kennen muss, um es benutzen zu können. Diese Sprache ist sehr inhaltsreich und wir schauen uns nur diejenigen Operationen an, die für das Lösen von Logeleien nötig sind.
Wir schreiben zusammen das Programm für *Mathematica*, um diese Aufgabe zu lösen.
Zu Beginn schreiben wir `Clear["`Global*`"]`, um alle zuvor im Programm gespeicherten Werte für Variablen zu löschen. Um eine gute Übersicht zu erschaffen, bezeichnen wir jede Bedingung mit b1, b2, b3 oder b4 usw. In unserem Fall ist `b1=Implies[a≠2&&b≠1,c==3]`, `b2=Implies[d≠4&&a≠1,b==3]`, `b3=c==4||d==2||a==3`, `b4=Implies[c≠1&&d≠1,b==3]`. Wir schreiben „=“ zwischen b1 und Aussagen, weil es sich hier um keine logische Operation handelt, sondern um eine eindeutige und klare Zuordnung.
Es gibt einige Bedingungen, die uns klar erscheinen, die aber für das Programm nicht eindeutig sind. Dass z.B. Aki nicht gleichzeitig der Hund und das Mädchen sein kann oder dass jeder Name einmal vorkommen muss, weiss Mathematica nicht. Dies müssen wir dem Programm mitteilen. Diese Bedingung lautet `b5=a≠b&&a≠c&&a≠d&&b≠c&&b≠d&&c≠d`.
Um zu überprüfen, ob wir hier alle Möglichkeiten aufgeschrieben haben, können wir das Kombinatorikverfahren gebrauchen: wenn wir n Buchstaben haben, können wir ohne Wiederholungen *(n-1)*n/2* Buchstabenpaare bilden. Wenn wir also 4 Buchstaben haben, können wir 3*4/2 = 6 Paare machen. Unser Resultat sollte somit stimmen.
Die letzte Bedingung b6 verknüpft alle vorherigen Bedingungen miteinander: `b6=And[b1,b2,b3,b4,b5]`. Wir könnten diesen Ausdruck auch als `b6=b1&&b2&&b3&&b4&&b5` aufschreiben, das erste ist aber einfacher.
Mathematica kann nicht unser Rätsel einfach lösen. Wir können aber das Programm dazu bringen, uns zu zeigen, welche Lösungen richtig und bzw. welche falsch sind. Wir lassen das Programm alle Möglichkeiten durchspielen und anschauen, welche von denen alle Bedingungen der Logelei erfüllen. Um dies zu erreichen, müssen wir dem Programm angeben, welche Werte unsere Variablen annehmen können. Das Programm spielt dann alle Möglichkeiten durch (z.B. a=2,b=3,c=1; d=4) und schaut, ob alle Bedingungen erfüllt werden. Wenn ja, dann ist dies eine der möglichen Lösungen. Wenn nicht, erscheint „false“.
Um nicht alle Möglichkeiten eintippen zu müssen, gebrauchen wir den Befehl `Table[`*`expr`*`,{n,min,max}]`, der Variablen verschiedene Werte annehmen lässt. Z.B. `Table[{a},{a,1,4}]` lässt die Variable a die Werte von 1 bis 4 annehmen. Es erscheint die Liste {{1},{2},{3},{4}}.
`Implies[..]`, `||` , `&&` und `Xor[..]` sind logische Operationen, die in *Mathematica* 2 mögliche Lösungen haben können: `true` oder `false`. Wenn wir wollen, dass das Programm nur die richtigen Lösungen anzeigt, können wir `If[`*`condition, t, f`*`]` gebrauchen. Anstatt „`condition`“ schreiben wir unsere Bedingung b6 hin. An Stelle von „`t`“ schreiben wir , was erscheinen soll, wenn eine Lösung richtig ist, und an Stelle von f, was erscheinen soll, wenn sie falsch ist. Wenn wir nun schreiben `If[b6,{a,b,c,d},""]`, so bedeutet dies, dass unsere ganze Bedingung b6 ist. Wenn eine mögliche Lösung b6 erfüllt, erscheint sie in der Form von {a,b,c,d}, wobei die Variablen die Werte dieser Lösung annehmen. Ist die mögliche Lösung falsch, erscheint eine leere Menge, anstatt der Variablen.
Jetzt lassen wir *Mathematica* so eine Liste aufschreiben:

```
Table[If[b6,{a,b,c,d},""],{a,1,4},{b,1,4},{c,1,4},{d,1,4}]
```

Die Variablen a, b, c und d können Werte von 1 bis 4 annehmen. Alle Möglichkeiten werden durchgespielt : {1,1,1,1},{1,1,1,2},{1,1,1,3} usw. Diese Beispiele werden dabei nicht

angezeigt, weil sie b6 nicht erfüllen. Sie erscheinen als {,,,}. Nur diejenige Lösungen, welche die Bedingung b6, bzw. (b1 und b2 und b3 und b4 und b5) erfüllen, erscheinen auf dem Bildschirm.
Ergänzung: Mathematica neigt dazu, nach jeder Eingabe eine Lösung oder eine Bestätigung zu geben, was manchmal den Übersicht erschwert. Um dies zu vermeiden, schreiben wir nach jeder Angabe ausser der letzten ein „ **;** “.

```
Clear["`Global*`"];

b1= Implies[a≠2&&b≠1,c==3];

b2= Implies[d≠4&&a≠1,b==3];

b3= Implies[c≠1&&d≠1,b==3];

b4= c==4||d==2||a==3;

b5= a≠b&&a≠c&&a≠d&&b≠c&&b≠d&&c≠d;

b6= And[b1,b2,b3,b4,b5];

Table[If[b6,{a,b,c,d},""],{a,1,4},{b,1,4},{c,1,4},{d,1,4}]
```

(Lösungen, die durchgespielt worden sind:)

```
{{{{,,,},{,,,},{,,,},{,,,}},{{,,,},{,,,},{,,,},{,,,}},{{,,,}
,{,,,},{,,,},{,,,}},{{,,,},{,,,},{,,,},{,,,}}},{{{,,,},{,,,}
,{,,,},{,,,}},{{,,,},{,,,},{,,,},{,,,}},{{,,,},{,,,},{,,,},
{{2,3,4,1},,,}},{{,,,},{,,,},{,,,},{,,,}}},{{{,,,},{,,,},{,,
,},{,,,}},{{,,,},{,,,},{,,,},{,,,}},{{,,,},{,,,},{,,,},{,,,
}},{{,,,},{,,,},{,,,},{,,,}}},{{{,,,},{,,,},{,,,},{,,,}},{{,
,,},{,,,},{,,,},{,,,}},{{,,,},{,,,},{,,,},{,,,}},{{,,,},{,,,
},{,,,},{,,,}}}}
```

Wir sehen, dass diese Aufgabe nur eine Lösung hat und zwar bei <u>a=2, b=3, c=4 und d=1</u>.
<u>Aki ist also der Junge, Bauzi ist der Hund, Knirps ist die Katze und Dicki ist das Mädchen.</u>

Wir schreiben alle Implies gemäss unserer Theorie um:

Implies[a≠2&&b≠1,c==3] ⇔ *(a==2||b==1)||c==3* ⇔ *a==2||b==1||c==3*

Implies[d≠4&&a≠1,b==3] ⇔ *(d==4||a==1)||b==3* ⇔ *d==4||a==1||b==3*

Implies[c≠1&&d≠1,b==3] ⇔ *(c==1||d==1)||b==3* ⇔ *c==1||d==1||b==3*

Wir schreiben die Bedingungen im Programm um und lassen es wieder alle Möglichkeiten durchspielen. Wir kommen wieder auf die gleiche Lösung, was darauf hinweist, dass unsere Theorie stimmen sollte.

3.1.2. Lösungsverfahren 1

Jetzt versuchen wir, diese Aufgabe ohne den PC mit Hilfe unserer Theorie zu lösen.
Das Lösungsverfahren 1 setzt kein logisches Denken voraus, sondern Fleiss und richtiges Einsetzen unserer Formeln:

(a==2||b==1||c==3)&&(d==4||a==1||b==3)&&(c==1||d==1||b==3)&&
(c==4||d==2||a==3)

⇔((a==2||b==1||c==3)&&(d==4||a==1||b==3))&&((c==1||d==1||b==3)&&
(c==4||d==2||a==3))

⇔ ((a==2&&d==4)||(a==2&&a==1) || (a==2&&b==3) || (b==1&&d==4)||
(b==1&&a==1) || (b==1&&b==3) || (c==3&&d==4) || (c==3&&a==1) ||
(c==3&&b==3)) && ((c==1&&c==4) || (c==1&&d==2)|| (c==1&&a==3)||
(d==1&&c==4) || (d==1&& d==2) || (d==1&&a==3) || (b==3&&c==4) ||
(b==3&&d==2) || (b==3&&a==3))

Wir sehen, dass einige Lösungen sich schon jetzt ausschliessen lassen, weil sie gegen unsere Regeln verstossen, wie zum Beispiel (`a==2&&a==1`). Wir wissen, dass unsere Variablen nicht gleichzeitig mehrere Werte annehmen können, darum ist diese Anordnung unmöglich.
Um nicht weiter allzu viel ausklammern zu müssen, können wir diese unmöglichen Lösungen bereits streichen. Alle oben grau geschriebene Lösungsmöglichkeiten fallen weg.

Unser Ziel ist es, die ganze Aufgabe in Form von A||B||C||D||... usw. zu bringen. Denn dann ist jede der Aussagen (A, B, C usw.) eine mögliche Lösung. Wir können zwischen ihnen die richtige Lösung finden, ohne viel überlegen zu müssen.

⇔((a==2&&d==4)|| (a==2&&b==3) || (b==1&&d==4) || (c==3&&d==4) ||
(c==3&&a==1)) && ((c==1&&d==2) ||(c==1&&a==3)|| (d==1&&c==4) ||
(d==1&&a==3) || (b==3&&c==4) || (b==3&&d==2))

⇔ (a==2&&d==4 && c==1&&d==2) || (a==2&&d==4 && c==1&&a==3)||
(a==2&&d==4 && d==1&&c==4) || (a==2&&d==4 && d==1&&a==3) ||
(a==2&&d==4 && b==3&&c==4) || (a==2&&d==4 && b==3&&d==2) ||
(a==2&&b==3 && c==1&&d==2) || (a==2&&b==3 && c==1&&a==3) ||
(a==2&&b==3 && d==1&&c==4) || (a==2&&b==3 && d==1&&a==3) ||
(a==2&&b==3 && b==3&&c==4) || (a==2&&b==3 && b==3&&d==2) ||
(b==1&&d==4 && c==1&&d==2) || (b==1&&d==4 && c==1&&a==3) ||
(b==1&&d==4 && d==1&&c==4) || (b==1&&d==4 && d==1&&a==3) ||
(b==1&&d==4 && b==3&&c==4) || (b==1&&d==4 && b==3&&d==2) ||
(c==3&&d==4 && c==1&&d==2) || (c==3&&d==4 && c==1&&a==3) ||
(c==3&&d==4 && d==1&&c==4) || (c==3&&d==4 && d==1&&a==3) ||
(c==3&&d==4 && b==3&&c==4) || (c==3&&d==4 && b==3&&d==2) ||
(c==3&&a==1 && c==1&&d==2) || (c==3&&a==1 && c==1&&a==3) ||
(c==3&&a==1 && d==1&&c==4) || (c==3&&a==1 && d==1&&a==3) ||
(c==3&&a==1 && b==3&&c==4) || (c==3&&a==1 && b==3&&d==2)

Lösungen, die unmöglich sind, sind grau gefärbt.
Blau gefärbt sind 2 Lösungen dieser Aufgabe. Wenn wir sie genau anschauen, können wir feststellen, dass sie gleich sind. D.h. a=2, b=3, c=4 und d=1.
Unser Resultat stimmt mit den Rechnungen in Mathematica überein.

3.1.3. Lösungsverfahren 2

Jetzt schauen wir uns die zweite mögliche Methode an.
Wir können zuerst alle Möglichkeiten aufschreiben und dann schrittweise die falschen ausschliessen oder umgekehrt diejenigen, die in Frage kommen, stehen lassen.
Wir schreiben 4! (Kombinatorikrechnen) = 4*3*2*1 = 24 Möglichkeiten auf. Wir wissen, dass wenn a=1 ist, dann kann weder b noch c noch d gleich 1 sein.
Wir zeichnen eine Tabelle. Wir erstellen (n+m) Spalten, wobei n die Anzahl der Variablen ist und m die Anzahl der Bedingungen, die zu diesen Variablen in der Aufgabe vorkommen. Hier haben wir 4 Variablen und 4 Bedingungen, also 8 Spalten. Die Anzahl der Zeilen ist gleich der Anzahl der möglichen Ergebnisse plus eins für die Beschriftung.
Wenn wir alle Möglichkeiten aufgeschrieben haben, lesen wir die erste Bedingung durch. Wir stellen Kreuze in der Spalte für diese Bedingung in denjenigen Zellen, in denen Lösungen vorkommen, welche die erste Bedingung erfüllen.
Wenn wir es durchgemacht haben, gehen wir zur zweiten Bedingung über. Wir brauchen aber nicht bei allen 24 Fällen zu überprüfen, ob sie die zweite Bedingung erfüllen, sondern nur bei denjenigen, die in der Spalte für die vorherige (erste) Bedingung angekreuzt worden sind. Denn nur wenn die möglichen Lösungen alle Bedingungen erfüllen, können sie das Resultat der Aufgabe sein. Und so gehen wir zur nächsten Bedingung über bis zum Schluss. In der Spalte der letzten Bedingung erhalten wir dann alle möglichen Lösungen der Logelei.

Tabelle zur Aufgabe 1 aus dem Buch „99 Logeleien von Zweistein“

a	b	c	d		b1	b2	b3	b4
1	2	3	4		x	x		
1	2	4	3					
1	3	2	4					
1	3	4	2					
1	4	2	3					
1	4	3	2		x	x		
2	1	3	4		x	x		
2	1	4	3		x			
2	3	1	4		x	x	x	
2	3	4	1		x	x	x	x
2	4	1	3		x			
2	4	3	1		x			
3	1	2	4		x	x		
3	1	4	2		x			
3	2	1	4					
3	2	4	1					
3	4	1	2					
3	4	2	1					
4	1	2	3		x			
4	1	3	2		x			
4	2	1	3					
4	2	3	1		x			
4	3	1	2					
4	3	2	1					

Kommentar: Wir haben : a, b, c und d; 1,2,3 und 4; jede Variable kann einen Wert von 1 bis 4 annehmen.

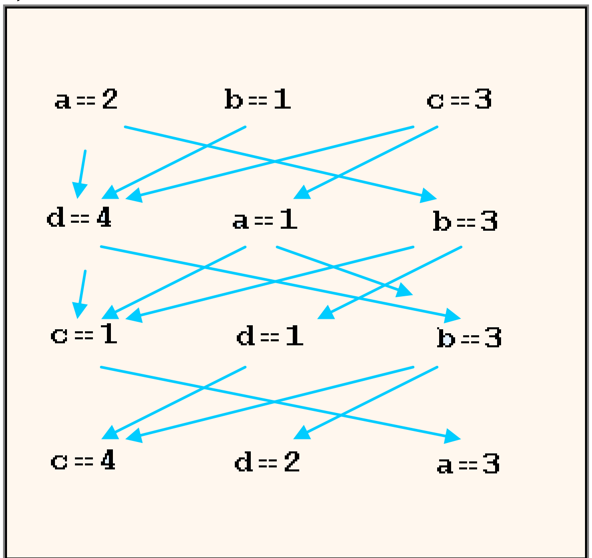
Bedingung 1 : `b1= a==2||b==1||c==3`
Bedingung 2 : `b2= d==4||a==1||b==3`
Bedingung 3 : `b3= c==1||d==1||b==3`
Bedingung 4 : `b4= c==4||d==2||a==3`

Wir kreuzen bei der ersten Bedingung in der Spalte b1 alle Lösungen mit `a==2`, mit `b==1` und mit `c==3` an. Bei der zweiten Bedingung schauen wir, welche von den angekreuzten möglichen Lösungen aus b1 `d==4, a==1` oder `b==3` enthalten, und kreuzen diese in der Spalte für die zweite Bedingung b2 an. Dann gehen wir zur dritten Bedingung über. Als mögliche Lösungen nehmen wir die angekreuzten Möglichkeiten aus der Spalte b2 an. Wir schauen wieder an, welche von ihnen `c==1,d==1 oder b==3` enthalten und kreuzen diese an. Wenn wir zur vierten Bedingung übergehen, nehmen wir Möglichkeiten aus der Spalte b3 und vergleichen sie mit der Bedingung 4. Diejenigen Möglichkeiten, die alle Bedingungen erfüllen, sind dann **alle** Lösungen.

Wir kommen auf das Resultat a=2, b=3, c=4 und d=1.

3.1.4. Lösungsverfahren 3

Dieses Verfahren ist mit dem ersten sehr verwandt, es schafft aber einen besseren Überblick. Wir schreiben auf der ersten Linie alle möglichen Lösungen der ersten Bedingung, auf der zweiten alle möglichen Lösungen der zweiten Bedingung usw. Wir verbinden die möglichen Lösungen der ersten Bedingung mit den möglichen Lösungen der zweiten, der zweiten mit der dritten und der dritten mit der vierten. Einige Lösungsmöglichkeiten können wir gleich ausschliessen, weil sie unmöglich sind, wie z.B. (a==2&&a==1). Bei diesem Verfahren können wir manchmal Glück haben und gleich beim ersten Versuch auf die Lösung kommen. Wir probieren aus, welcher der Wege zur Lösung führt.



Lösungsverfahren 3 Logelei 1

Fangen wir bei a==2 an: (a==2&&d==4) || (a==2&&b==3). Aus der dritten Bedingung folgt: (a==2&&d==4&&c==1) || (a==2&&d==4&&b==3) || (a==2&&b==3&&c==1) || (a==2&&b==3&&d==1). Die letzte erfüllt die vierte Bedingung: (a==2&&b==3&&d==1&&c==4). Dies ist die Lösung.
Fangen wir bei b==1 an, dann haben wir bei der zweiten Bedingung (b==1&&d==4); bei der dritten halten wir an: Es gibt keinen möglichen Weg zur vierten Bedingung.
Wenn wir mit c==3 anfangen, haben wir nach der zweiten Bedingung: (c==3&&d==4) || (c==3&&a==1). Es gibt in der dritten Bedingung keine Aussagen, die mindestens eine der beiden erfüllen würden. Darum gibt es hier keine Lösung.

3.1.5. Lösungsverfahren 4

Dieses Verfahren habe ich entdeckt, als ich ausprobiert habe, Logeleien mit einer einfachen Tabelle zu lösen. Es ist nicht bei allen Logeleien bequem zum Anwenden, aber für die einfacheren Logeleien eignet sich dieses Verfahren sehr gut und geht am schnellsten.
Wir zeichnen eine Tabelle, in der horizontal Variablen und vertikal ihre möglichen Werte vorkommen.
Für dieses Verfahren brauchen wir Bedingungen der Form (..==..)||(..==..)||(..). Wir können jede der Bedingung in diese Form bringen, indem wir unsere Formeln anwenden.

In unsrem Beispiel haben wir 4 Bedingungen, die wir bereits in diese Form gebraucht haben. Aus der ersten Bedingung sehen wir, dass eine von den drei Aussagen a=2, b=1 und c=3 unbedingt zur Lösung gehört. Wir können also in der Tabelle unter a=2, b=1 und c=3 je ein „b1" schreiben. Genau das gleiche machen wir mit den anderen 3 Bedingungen (b2, b3, b4).
Jetzt haben wir b1, b2, b3 und b4 in der Tabelle. Da diese Bedingungen einander ergänzen und einander nicht ausschliessen dürfen, muss **jede der Bedingungen mindestens einmal** in der Lösung vorkommen.

Nehmen wir zuerst die erste Bedingung b1 in der Tabelle bei b=1. Wenn also b=1 ist, kann keine andere Variable mehr gleich 1 sein. Also müssen die zweite, die dritte und die vierte Bedingung in den Zeilen für die Zahlen 2, 3, 4 bei a, c oder d liegen. Es bleibt nur eine b2: bei d=4. Jetzt suchen wir nach b3. Diese b3 muss sich irgendwo in den Zellen a=2, a=3, c=2 oder c=3 befinden. Wir finden aber keine. Also ist b ungleich 1. Somit bleiben uns noch 2 andere Möglichkeiten aus der ersten Bedingung zum Ausprobieren.
Wir nehmen b1 bei c=3. Nun können wir alle c und 3 in der Tabelle ausschliessen. Jetzt schauen wir, was noch übrig bleibt: zwei b2, eine b3 und eine b4. Am günstigsten nehmen wir das, was einmal vorkommt. Es spielt keine Rolle, in welcher Reihenfolge wir vorgehen, schlussendlich müssen einfach alle „Bedingungen" dabei sein und unseren Regeln entsprechen. Wir nehmen jetzt b3. Und wir sehen, dass wir schon jetzt keine b4 mehr haben können. Darum stimmt dieser Weg nicht.
b1 soll also bei a=2 sein. Wir decken alle a und 2 ab. Uns bleiben zwei b2, drei b3 und eine b4. Also nehmen wir zuerst diese verbliebene b4, da sie nur einmal vorkommt. Wir decken wieder ab, diesmal alle c und 4. Es bleibt nur eine b2 übrig, und zwar bei b=3. Da sich bei b=3 sogar zwei Bedingungen befinden, können wir beide zählen, da sie nur das Gleiche aussagen, einander also nicht ausschliessen. Wir haben: b1 bei a=2, b2 und b3 bei b=3, b4 bei c=4; d kann dann nur bei 1 liegen.
Wir können auch feststellen, dass b3 bei d=1 überflüssig ist. Denn wir sind auch ohne diese Angabe auf das einzige richtige Resultat gekommen.

```
b1= a==2||b==1||c==3
b2= d==4||a==1||b==3
b3= c==1||d==1||b==3
b4= c==4||d==2||a==3
```

Tabelle zum Lösungsverfahren 4

Wert / Variable	a	b	c	d
1	b2	b1	b3	b3
2	b1			b4
3	b4	b2,b3	b1	
4			b4	b2

Das Vorgehen bei der Annahme b1 sei bei b==1

a)b==1 [b1]

Wert / Variable	a	b	c	d
1		b1		
2	b1			b4
3	b4		b1	
4			b4	b2

b) nur eine b2 bei d==4 ⇒ keine b3 mehr ⇒ keine Lösung

Wert / Variable	a	b	c	d
1		b1		
2	b1			
3	b4		b1	
4				b2

Das Vorgehen bei der Annahme b1 sei bei a==2

a) a==2 [b1]

Wert / Variable	a	b	c	d
1		b1	b3	b3
2	b1			
3		b2,b3	b1	
4			b4	b2

b) nur eine b4 bei c==4

Wert / Variable	a	b	c	d
1		b1		b3
2	b1			
3		b2,b3		
4			b4	

c) Wir haben nur noch eine b2 bei b==3. b3 befindet sich auch in dieser Zelle, also, wenn wir b==3 nehmen, haben wir alle „Bedingungen“. Also ist die Lösung <u>a=2, c=4, b=3 und d=1.</u>

3.2- 3.6. Lösen von Logeleien (Fortsetzung)

Wir schauen uns ein paar andere Logeleien an. Wir lösen sie mit Hilfe einiger unserer Lösungsverfahren und schauen uns auch an, was für speziellen alternativen Lösungsverfahren sie haben könnten. Wir werden sehen, dass es am besten funktioniert, wenn wir mehrere Methoden gleichzeitig für das Lösen von einer Logelei anwenden.
Die Logelei 94, welche uns am Anfang der Arbeit gestellt wurde, wird am Schluss dieses Kapitels gelöst.
Die vollständigen und erweiterten Lösungen für das Programm *Mathematic*a , das Lösungsverfahren 1 und einige andere Methoden zu den unten folgenden Beispiele sind im Anhang auf der CD zu finden.

3.2. Aufgabe 3 im Buch „99 Logeleien von Zweistein“

„ Meiers werden uns heute Abend besuchen “, kündigt Herr Müller an. „ Die ganze Familie, also Herr und Frau Meier, nebst ihren drei Söhnen Tim, Kay und Uwe? “, fragt Frau Müller bestürzt. Darauf Herr Müller, der keine Gelegenheit vorübergehen lässt, seine Frau zum logischen Denken anzureizen: „ Nein, ich will es dir so erklären: Wenn Vater Meier kommt, dann bringt er auch seine Frau mit. Mindestens einer der beiden Söhne Uwe und Kay kommt. Entweder kommt Frau Meier oder Tim. Entweder kommen Tim und Kay oder beide nicht. Und wenn Uwe kommt, dann auch Kay und Herr Meier. So, jetzt weisst, wer uns heute abend besuchen wird. “

Wir haben 5 Personen und 2 Eigenschaften („kommt“ - „kommt nicht“). Die Anzahl der Eigenschaften ist kleiner, dann müssen sie Zahlen sein. Also:
Herr Meier sei *h*, Frau Meier sei *f*, Tim sei *t* und Kay sei *k* und Uwe sei *u*;
„Kommt“ sei *1* und „kommt nicht“ *2*.
Personen werden Eigenschaften zugeordnet, d.h. wir können „gleich“ gebrauchen.

„Wenn Vater Meier kommt, dann bringt er auch seine Frau mit.“ $\Rightarrow$ *Implies[h==1,f==1]* $\Leftrightarrow$ *(h≠1||f==1)* $\Leftrightarrow$ ***(h==2||f==1)***

„Mindestens einer der beiden Söhne Uwe und Kay kommt.“ $\Rightarrow$ ***(u==1||k==1)***

„Entweder kommt Frau Meier oder Tim.“ $\Rightarrow$ *Xor[f==1,t==1]* $\Leftrightarrow$ *((f==1&&t≠1)||(f≠1&&t==1))* $\Leftrightarrow$ ***((f==1&&t==2)||(f==2&&t==1))***

„Entweder kommen Tim und Kay oder beide nicht.“ $\Rightarrow$ *Xor[t==1&&k==1,t≠1&&k≠1]* $\Leftrightarrow$ ***((t==1&&k==1)||(t==2&&k==2))***

„Und wenn Uwe kommt, dann auch Kay und Herr Meier.“ $\Rightarrow$ *Implies[u==1,k==1&&h==1]* $\Leftrightarrow$ *((u≠1)||(k==1&&h==1)* $\Leftrightarrow$ ***((u==2)||(k==1&&h==1)***

(Lösung: h=2,f=2,t=1,k=1,u=2. Nur Tim und Kay kommen)

Nach der Umformung dieser Logelei haben wir viele eingeklammerte „Oder“ zwischen „Und“, darum ist hier das erste Verfahren schlecht zum Anwenden.
Wir haben 5 Variablen, die je einen von zwei Werten annehmen können. Die Wiederholungen sind vorhanden. Es gibt also 2^5= 32 mögliche Lösungen. So eine Tabelle zu zeichnen wäre auch nicht die beste Lösung.
Das dritte Verfahren könnte aber gehen. Wir schreiben die Bedingungen wie in der ersten Logelei auf. Dabei ändern wir die Reihenfolge der Bedingungen so, dass die gleichen Buchstaben möglichst gleich untereinander stehen.

Wir fangen mit dem Weg u==1 an: k==1&&k==2 → unmöglich. Der Weg k==1&&u==2&&h==2&&f==1&&t==2 : k==1&&k==2 → unmöglich.

Der Weg k==1&&u==2&&h==2&&f==2&&t==1 → eine Lösung. Der Weg k==1&&u==2&&f==1: k==1&&k==2 → unmöglich. Der Weg k==1&&k==1&&h==1&&f==1: k==1&&k==2 → unmöglich.

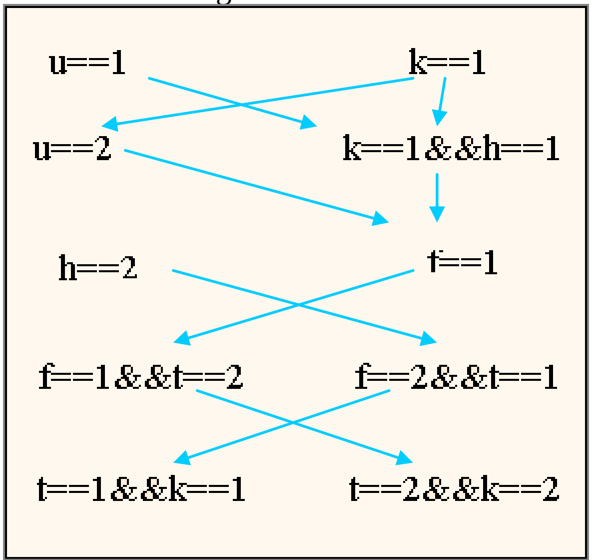


Abbildung zum Lösungsverfahren 3

Das vierte Lösungsverfahren würde auch gehen. Denn es gibt nicht allzu viele Bedingungen und jede Bedingung besteht aus der Aussagen, welche „==“ enthalten (nicht „≠“) und „Oder“.

```
b1= h==2||f==1
b2= u==1||k==1
b3= (f==1&&t==2)||(f==2&&t==1)
b4= (t==1&&k==1)||(t==2&&k==2)
b5= u==2||(k==1&&h==1)
```

In manchen Bedingungen treffen wir die Aussagen der Art (A&&B)||(C&&D), so wie b3. Da können wir nicht nur eine „Bedingung“ z.B. b3 bei der Aussage A schreiben, denn wir müssen auch B berücksichtigen. Wir teilen b3 in zwei Aussagen: b3a und b3b. b3a ist dann A&&B und b3b - C&&D. b3a oder b3b sind nur dann gültig, wenn sie je zwei mal vorkommen, also können wir nur 2*b3a oder 2*b3b als eine b3 bezeichnen.
Unsere Variablen können einander gleich sein, aber eine Variable kann nicht zwei Werte annehmen. So wenn wir eine Anordnung aus den Aussagen annehmen, dürfen wir nicht alle Variablen für diesen Wert ausschliessen, wir müssen aber für diese Variable die anderen Werte ausschliessen.

Tabelle zum Lösungsverfahren 4

Wert / Variable	h	f	t	k	u
1	b5b	b1, b3a	b3b, b4a	b2, b4a, b5b	b2
2	b1	b3b	b3a, b4b	b4b	b5

Wir haben zwei b1 und zwei b5, darum fangen wir mit ihnen an.
Nehmen wir an, b1 sei bei f=1, dann bleiben uns beide b5b, beide b4a, beide b4b, zwei b2, nur eine b3a und nur eine b3b. Wir können schon jetzt aufhören, weil es uns die anderen Hälften der b3a und b3b fehlen.
Dann muss b1 bei h=2 liegen. b5b ist somit ausgeschlossen, da eine Hälfte dieser „Bedingung“ in der Zelle h=1 liegt. Dann bleibt uns für die fünfte Bedingung b5. Wir haben also b1 und b5. Es fehlen uns b2, b3 und b4. Es bleibt uns nur eine b2 bei k=1 übrig, darum nehmen wir sie. So haben wir jetzt b1, b2, eine b4a und b5. Es bleiben uns beide b3a, beide b3b, aber nur eine b4a und eine b4b. Wir haben bereits eine b4a bei k=1, so müssen wir die zweite b4a nehmen, damit es aufgeht. Mit b4a bei t=1 haben wir auch b3b- eine Hälfte der letzten fehlenden „Bedingung“, die zur Lösung führt..

3.3. Aufgabe 4 im Buch „99 Logeleien von Zweistein“

„Jeder der drei Freunde Anton, Hans und Karl spielt zwei Musikinstrumente, man kann also jedem zwei der folgenden Bezeichnungen geben: Flötist, Trommler, Geiger, Cellist, Trompeter und Pianist.
Der Florist verulkt den Geiger gern; der Trompeter und der Geiger gehen mit Anton gelegentlich zum Fussball; der Cellist hat beim Trommler Schulden; der Florist mit der Schwester des Cellisten verlobt; Hans hat dem Trompeter das Instrument versteckt; und im Skat hat Karl gegen Hans und den Cellisten gewonnen.
So, nun dürfte es klar sein, wer welche Instrumente spielt. „

Wir haben 3 Namen und 6 Musikanten. Mit Variablen bezeichnen wir dann Instrumente, die Musikanten spielen und mit Zahlen ihre Namen.
Anton sei *1*, Hans sei *2* und Karl sei *3*;
Flötist sei *a*, Trommler *b*, Geiger *c*, Cellist *d*, Trompeter *e* und Pianist sei *f*.

„Der Flötist verulkt den Geiger gern. “ $\Rightarrow$ $\mathbf{a \neq c}$

„Der Trompeter und der Geiger gehen mit Anton gelegentlich zum Fussball.“ Was bedeutet, dass es sich hier um 3 verschiedene Personen handelt $\Rightarrow$ ***e ≠ c && e ≠ 1 && c ≠ 1***

„Der Cellist hat beim Trommler Schulden.“ $\Rightarrow$ ***d ≠ b***

„Der Flötist mit der Schwester des Cellisten verlobt.“ $\Rightarrow$ ***a ≠ d***

„Hans hat dem Trompeter das Instrument versteckt.“ $\Rightarrow$ ***2 ≠ e***

„Im Skat hat Karl gegen Hans und den Cellisten gewonnen.“ $\Rightarrow$ ***d ≠ 2 && 3≠ d***

(Lösung: a=3, b=2, c=2, d=1, e=3, f=1. Anton ist Cellist und Pianist, Hans ist Trommler und Geiger, Karl ist Flötist und Trompeter.)

So haben wir jetzt 6 Aussagen, die wir mir durch „und“ miteinander verknüpfen können. Es kommen keine „Oder“ dazwischen, was darauf hinweist, dass wir hier schon eine fertige Lösung haben. Wir müssen sie nur entschlüsseln können. Dabei brauchen wir keine der oben beigebrachten Methoden zu verwenden.
In solchem Fall gehen wir folgendermassen vor: Wir schreiben alle Bedingungen, die wir haben, zusammen und ändern die Reihenfolge so, dass die gleichen Variablen in verschiedenen Aussagen gleich nacheinander folgen. (Wir könnten beispielsweise „a≠c“ in die Form (a==b)||(a==d)||(a==e)||(a==f) bringen und dann ausklammern. Wir verzichten aber auf diesen Aufwand.)

a ≠ c && e ≠ c && e ≠ 1 && c ≠ 1 && d ≠ b && a ≠ d && 2 ≠ e && d ≠ 2 && 3≠d ⇔

a ≠ c && a ≠ d && c ≠ 1 && d ≠ b && ***d ≠ 2*** *&&* ***d≠3*** *&&* ***e ≠ 1*** *&&* ***e ≠ 2*** *&& e ≠ c* ⇔

a ≠ c && a ≠ **d** *&& c ≠ 1 &&* **d** *≠ b &&* **d==1** *&&* **e==3** *&& e ≠ c* ⇔

a ≠ c && a ≠ 1 && c ≠ 1 && 3 ≠ c && 1≠b && **d==1** *&&* **e==3** ⇔

a ≠ **2** *&& a ≠ 1 &&* **c==2** *&& 1≠b &&* **d==1** *&&* **e==3** ⇔

a ==3 *&& c==2 &&* **b==2** *&& d==1 &&* **e==3** (jede Zahl kann zwei Variablen gleich sein)
⇔ ***e==3 && d==1 && c==2 && a==3 && b==2 && f==1***

3.4. Aufgabe 9 im Buch „99 Logeleien von Zweistein“

„Von den drei Herren Amann, Bemann und Cemann hat einer blondes, einer braunes und einer schwarzes Haar; einer hat einen Schnauzbart, einer einen Backenbart und einer hat einen Spitzbart.
Cemann hat weder schwarzes Haar noch einen Backenbart.
Der blonde Herr ist weder Bemann, noch hat er einen Schnauzbart.
Wenn der Mann mit dem Schnauzbart entweder Bemann oder Cemann ist, dann hat der Spitzbärtige schwarzes Haar.

Wenn der Herr mit dem schwarzen Haar Bemann ist, dann hat der mit dem braunen Haar keinen Schnauzbart.
Wie sehen die Herren Amann, Bemann und Cemann aus? „

Wir haben 3 Namen und 6 Eigenschaften. Mit Variablen bezeichnen wir Eigenschaften und mit Zahlen Namen.
Amann sei *1*, Bemann sei *2* und Cemann sei *3*;
Schwarzes Haar sei *a*, blondes *b*, braunes *c*, Schnauzbart sei *d*, Backenbart *e* und Spitzbart *f*.

„Cemann hat weder schwarzes Haar noch einen Backenbart.“ ⇒ ***3 ≠ a && 3≠e***

„Der blonde Herr ist weder Bemann, noch hat er einen Schnauzbart.“ ⇒ ***b ≠ 2 && b ≠ d***

„Wenn der Mann mit dem Schnauzbart entweder Bemann oder Cemann ist, dann hat der Spitzbärtige schwarzes Haar.“ Wir haben „entweder- oder“ in der Bedingung, aber wir können einfach „oder“ nehmen, weil wir wissen, dass wenn Bemann Schnauzbart hat, dann hat Cemann automatisch keinen mehr und umgekehrt. ⇒ *Implies[d==2||d==3, f==a]* ⇔ ***(d≠2&&d≠3)||(f==a)***

„Wenn der Herr mit dem schwarzen Haar Bemann ist, dann hat der mit dem braunen Haar keinen Schnauzbart.“ ⇒ *Implies[a==2, c≠d]* ⇔ ***(a≠2||c≠d)***

(Lösung: a=1, b=3, c=2, d=1, e=2, f=3. Amann hat schwarzes Haar und Schnauzbart, Bemann hat braunes Haar und Backenbart, Cemann hat blondes Haar und Spitzbart.)

Wir haben zwei „Oder“ in diesem Beispiel, darum verwenden wir das erste Verfahren:

```
⇔3≠a&&3≠e && b≠2&&b≠d &&(d≠2&&d≠3)||f==a))&&(a≠2||c≠d)⇔
((3≠a&&3≠e&&b≠2&&b≠d&& d≠2&&d≠3)||
(3≠a&&3≠e&&b≠2&&b≠d&&f==a))&&(a≠2||c≠d)⇔
(3≠a&&3≠e&&b≠2&&b≠d&& d≠2&&d≠3&& a≠2)||
(3≠a&&3≠e&&b≠2&&b≠d&& d≠2&&d≠3&& c≠d)||
(3≠a&&3≠e&&b≠2&&b≠d&&f==a&& a≠2)||
(3≠a&&3≠e&&b≠2&&b≠d&&f==a&& c≠d)
```

Wir haben 4 mögliche Lösungen, bei denen wir nicht gleich sehen können, welche stimmt und welche nicht. Dafür verwenden wir unsere Methode wie in der Logelei 4:

a) 3≠a&&3≠e&&b≠2&&b≠d && d≠2&&d≠3&&a≠2 ⇔
3≠a&&a≠2&&b≠2&&b≠d &&d≠2&&d≠3&&3≠e ⇔
a==1&&b≠2&&b≠1 &&**d==1**&&3≠e ⇔
a==1&&b==3 &&d==1&&e==2 *(jede Zahl kann nur zwei Variablen gleich sein)* ⇔
(a, b und c müssen Werte von 1 bis 3 annehmen bzw. d, e und f auch)
a==1&&b==3&&c==2 && d==1&&e==2&&f==3
⇒ Lösung 1

b) $3 \neq a \&\& 3 \neq e \&\& b \neq 2 \&\& b \neq d \&\& d \neq 2 \&\& d \neq 3 \&\& c \neq d \Leftrightarrow$

$3 \neq a \&\& b \neq 2 \&\& b \neq d \&\& c \neq d \quad \&\& d \neq 2 \&\& d \neq 3 \&\& 3 \neq e \Leftrightarrow$

$3 \neq a \&\& b \neq 2 \&\& b \neq 1 \&\& c \neq 1 \quad \&\& d == 1 \&\& 3 \neq e \Leftrightarrow$

$a == 1 \&\& b == 3 \&\& c == 2 \quad \&\& d == 1 \&\& e == 2 \&\& f == 3$

$\Rightarrow$ <u>Lösung 2</u> (gleich der ersten Lösung)

c) $3 \neq a \&\& 3 \neq e \&\& b \neq 2 \&\& b \neq d \&\& f == a \&\& a \neq 2 \Leftrightarrow$

$a == 1 \&\& 3 \neq e \&\& b \neq 2 \&\& b \neq d \&\& f == 1 \Leftrightarrow$

$a == 1 \&\& b == 3 \&\& c == 2 \quad \&\& 3 \neq d \&\& 3 \neq e \&\& f == 1 \Leftrightarrow$

$a == 1 \&\& b == 3 \&\& c == 2 \quad \&\& d == 2 \&\& e \neq 3 \&\& f == 1$

$\Rightarrow$ <u>keine Lösung</u> (e≠3 *und d==2&&f==1 können nicht gleichzeitig miteinander existieren*)

d) $3 \neq a \&\& 3 \neq e \&\& b \neq 2 \&\& \mathbf{b \neq d} \&\& f == a \&\& \mathbf{c \neq d}$

$\Rightarrow$ *<u>keine Lösung</u> (d==a und f==a können nicht gleichzeitig miteinander existieren)*

3.5. Aufgabe 20 im Buch „99 Logeleien von Zweistein“

„Wenn sich Frauen über ihr Alter unterhalten, darf man nicht erwarten, dass sie stets ehrlich sind. In der folgenden Konversation zwischen Anna, Barbara und Christa sagt jede der drei Damen zweimal die Wahrheit und einmal die Unwahrheit.
Anna: „Barbara ist zwei Jahre älter als ich.“
Barbara: “Christa ist 32 Jahre alt.“
Christa: “Anna ist älter als ich.“
Barbara: „Der Altersunterschied zwischen Christa und mir beträgt drei Jahre.“
Christa: “Anna ist 30 Jahre alt.“
Anna: “Ich bin 29 Jahre alt.“
Barbara: “Mindestens eine von euch ist jünger als ich.“
Anna: “Ich bin ein Jahr älter als Christa.“
Christa: “Anna ist drei Jahre jünger als Barbara.“
Wie alt sind Anna, Barbara und Christa? „

Wir haben 3 Namen und 3 Alter. Mit Variablen bezeichnen wir Frauen und mit Zahlen ihr Alter, das in der Aufgabe vorkommt. Jeder Frau kann genau ein Alter zugeordnet werden, wir gebrauchen also „gleich“.
Anna sei a, Barbara b und Christa c. Das Alter bezeichnen wir genau mit denjenigen Zahlen, die in der Logelei vorkommen, d.h. 29, 30, 32 usw.
Wir wissen auch, dass jede der Frauen die Unwahrheit sagt. Wir bezeichnen die Aussagen jeder der Frauen mit verschiedenen Buchstaben: für Anna nehmen wir A1, A2 und A3, für Barbara B1, B2 und B3, für Christa C1, C2 und C3 , und stellen die Bedingung, die genau zwei der drei Aussagen jeder Frau stimmen lässt.

„Anna: „Barbara ist zwei Jahre älter als ich.““ Es sagt Anna, also nehmen wir A1. Sie redet von der Barbara und sich selber, sie stellt eine Gleichung auf: b==a+2. Wir wissen nicht, ob diese Gleichung genau stimmt, darum gebrauchen wir „==“. ⇒ ***A1= b==a+2***

„Barbara: “Christa ist 32 Jahre alt.““ ⇒ ***B1= c==32***

„Christa: “Anna ist älter als ich.““ ⇒ ***C1= a>c***

„Barbara: „Der Altersunterschied zwischen Christa und mir beträgt drei Jahre."“ Wir wissen nicht wer älter ist, darum können wir „oder“ gebrauchen und zwei verschiedene Differenzen aufschreiben. ⇒ ***B2= (c-b==3)||(b-c==3)***

„Christa: “Anna ist 30 Jahre alt."“ ⇒ ***C2= a==30***

„Anna: “Ich bin 29 Jahre alt."“ ⇒ **A2= a==29**

„Barbara: “Mindestens eine von euch ist jünger als ich."“ Da gebrauchen wir wieder „oder“ weil wir nicht wissen, wer jünger ist. ⇒ **B3= c<b||a<b**

„Anna: “Ich bin ein Jahr älter als Christa."“ ⇒ ***A3= a==c+1***

„Christa: “Anna ist drei Jahre jünger als Barbara."“ ⇒ ***C3= a==3+b***

Es ist möglich, diese Logelei mit Hilfe des Lösungsverfahrens 1 zu lösen, da es aber wegen der grossen Anzahl von „Oder“ nicht die beste Lösung ist, greifen wir zu einer anderen Methode. Das erste Verfahren zu dieser Logelei ist aber im Anhang zu finden.

Hier versuchen wir, diese Aufgabe genau zu betrachten und logische Schlüsse zu ziehen.
Wir haben 9 Bedingungen:

```
A1= b==a+2;
A2= a==29;
A3= a==c+1;
B1= c==32;
B2= (c-b==3)||(b-c==3);
B3= c<b||a<b;
C1= a>c;
C2= a==30;
C3= a==3+b;
```

Wenn wir sie vergleichen, merken wir:
A2 und C2 können nicht miteinander existieren, also A2 oder C2 ist falsch;
A1 und C3 können auch nicht gleichzeitig wahr sein, also muss eines von beiden falsch sein.
Wir können jetzt sagen: entweder (A2 **sowie** C3) sind falsch oder (A1 **sowie** C2).
Betrachten wir zuerst den Fall, wenn A1 und C2 falsch sind. A2, A3, C1 und C3 sind dann wahr. Also, a=29, c=28, 29>28, b=26. Jetzt schauen wir uns die Aussagen der Barabara an: B1 stimmt nicht, B2 stimmt nicht, B3 stimmt auch nicht. Also, dieser Weg „A1- falsch und C2- falsch“ ist falsch.
Uns bleibt der Weg „A2- falsch und C3- falsch“. Dann: a=30, b=32, c=29, 30>29,. Was sagt Barbara dazu? B1 (c=32) stimmt nicht, dann sollten die anderen zwei Aussagen stimmen. 32-29=3, 32>30. B2 und B3 stimmen
⇒ <u>Lösung</u>: <u>Aussagen A2, B1 und C3 stimmen nicht,</u>
<u>Anna ist 30 Jahre alt, Barbara ist 32 Jahre alt und Christa ist 29 Jahre alt.</u>

3.6. Aufgabe 94 im Buch „99 Logeleien von Zweistein“

„Die Herren Amann, Bemann, Cemann und Demann heissen – nicht unbedingt in derselben Reihenfolge - mit Vornamen Erich, Fritz, Gustav und Heinrich. Sie sind alle verheiratet. Ausserdem weiss man über sie und ihre Ehefrauen noch dies: Entweder ist Amanns Vorname Heinrich, oder Bemann Frau heisst Inge. Wenn Cemann mit Josefa verheiratet ist, dann – und nur in diesem Falle – heisst Klaras Mann nicht Fritz.
Wenn Josefas Mann nicht Erich heisst, dann ist Inge mit Fritz verheiratet.
Wenn Luises Mann Fritz heisst, dann ist der Vorname von Klaras Mann nicht Gustav.
Wenn die Frau von Fritz Inge heisst, dann ist Erich nicht mit Josefa verheiratet.
Wenn Fritz nicht mit Luise verheiratet ist, dann heisst Gustavs Frau Klara.
Entweder ist Demann mit Luise verheiratet, oder Cemann heisst Gustav.
Wie heissen die Herren mit vollem Namen, wie ihre Ehefrauen mit Rufnamen? „

Wir haben 4 Herren, 4 ihre Frauen und 4 ihre Vornamen. Wir nehmen Zahlen von 1 bis 4 für Herren und Variablen von e bis l für Frauen- und Herrenvornamen.
1 sei Amann, 2 sei Bemann, 3- Cemann und 4- Demann.
Erich sei e, Fritz sei f, Gustav g, Heinrich h, Inge i, Josefa j, Klara k und Luise l.

„Entweder ist Amanns Vorname Heinrich, oder Bemann Frau heisst Inge.“
$\Rightarrow Xor[1==h,2==i] \Leftrightarrow \mathbf{((1==h\&\&2\neq i)||(1\neq h\&\&2==i))}$

„Wenn Cemann mit Josefa verheiratet ist, dann – und nur in diesem Falle – heisst Klaras Mann nicht Fritz.“ Das heisst auch, dass wenn Cemann mit Josefa nicht verheiratet ist, dann heisst Klaras Mann Fritz. $\Rightarrow Implies[3==j,k\neq f]\&\& \; Implies[3\neq j,k==f] \Leftrightarrow (3\neq j||k\neq f)\&\&(3==j||k==f) \Leftrightarrow \mathbf{(3\neq j\&\&k==f)||(k\neq f\&\&3==j)}$

„Wenn Josefas Mann nicht Erich heisst, dann ist Inge mit Fritz verheiratet.“
$\Rightarrow Implies[j\neq e,i==f] \Leftrightarrow \mathbf{(j==e||i==f)}$

„Wenn Luises Mann Fritz heisst, dann ist der Vorname von Klaras Mann nicht Gustav.“
$\Rightarrow Implies[l==f,k\neq g] \Leftrightarrow \mathbf{(l\neq f||k==g)}$

„Wenn die Frau von Fritz Inge heisst, dann ist Erich nicht mit Josefa verheiratet.“
$\Rightarrow Implies[i==f,e\neq j] \Leftrightarrow \mathbf{(i\neq f||e\neq j)}$

„Wenn Fritz nicht mit Luise verheiratet ist, dann heisst Gustavs Frau Klara.“
$\Rightarrow Implies[f\neq l,g==k] \Leftrightarrow \mathbf{(f==l||g\neq k)}$

„Entweder ist Demann mit Luise verheiratet, oder Cemann heisst Gustav.“
$\Rightarrow Xor[4==l,3==g] \Leftrightarrow \mathbf{((4\neq l\&\&3==g)||(4==l\&\&3\neq g))}$

(Lösung: e=4, f=2, g=1, h=3, i= 2, j=3, k=1, l=4)

Um uns für eine oder mehrere Methoden zu entscheiden, schreiben wir zuerst die ganze Aufgabe auf: $((1==h\&\&2\neq i)||(1\neq h\&\&2==i))\&\& ((3\neq j\&\&k==f)||(k\neq f\&\&3==j))\&\& (j==e||i==f)\&\& (l\neq f||k==g)\&\& (i\neq f||e\neq j)\&\& (f==l||g\neq k)\&\& ((4\neq l\&\&3==g)||(4==l\&\&3\neq g)) \Leftrightarrow$

$((1==h\&\&2\neq i)||(1\neq h\&\&2==i))\&\&((3\neq j\&\&k==f)||(k\neq f\&\&3==j))\&\& (j==e||i==f)\&\& (i\neq f||e\neq j)\&\& (l\neq f||k==g)\&\& (f==l||g\neq k)\&\& ((4\neq l\&\&3==g)||(4==l\&\&3\neq g)) \Leftrightarrow$

$((1==h\&\&2\neq i)||(1\neq h\&\&2==i))\&\& ((3\neq j\&\&k==f)||(k\neq f\&\&3==j))\&\& ((j==e\&\&i\neq f)|| (i==f\&\&e\neq j)) \&\& ((l\neq f\&\&k==g)||(f==l\&\&g\neq k)) \&\& ((4\neq l\&\&3==g)||(4==l\&\&3\neq g))$

Es wird schwierig sein, alle fünf „Oder" auszuklammern, darum greifen wir zum Lösungsverfahren 3. Wir schreiben die Bedingungen mit den gleichen Buchstaben untereinander. Als Nächstes klammern wir die erste Bedingung mit der zweiten und die dritte mit der vierten aus. Nun stellen wir ein neues Schema dar.

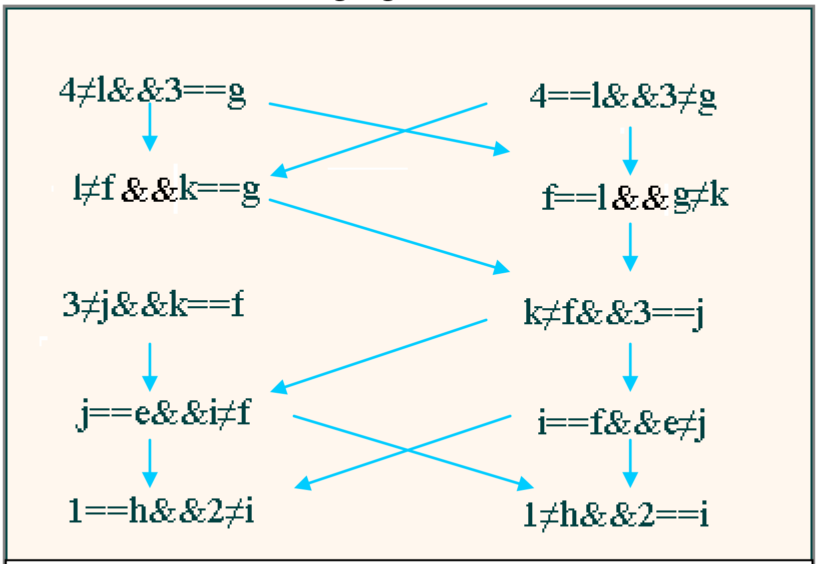


Abbildung zum Lösungsverfahren 3, Schritt 1

Wir schauen uns an, welche der Aussagen sich offensichtlich ausschliessen lassen. Wie z.B. *(4≠l&&3==g&&l≠f&& k==g)* && *(k≠f&&3==j &&j==e&&i≠f)*: 3==g&&3==j&&g==k ist eine unmögliche Lösung, weil nur eine Variable von e bis h sowie auch eine Variable von i bis l gleich 3 sein kann. Wir zeichnen nur dort Pfeilen ein, wo mögliche Lösungen vorhanden sind.

Auf diesem Schema haben wir noch 5 mögliche Wege, die sich nicht offensichtlich ausschliessen lassen, die wir aber genau betrachten können, um zur Lösung zu gelangen. Wir schreiben sie auf:

a) *4≠l&& 3==g&& f==l&&g≠k && k≠f && 3==j && j==e &&i≠f;*
b) *4≠l&& 3==g&& l≠f&& k==g && k≠f && 3==j && j≠e &&i==f;*
c) *4==l&& 3≠g && f==l&&g≠k && k≠f &&3==j &&j==e &&i≠f;*
d) *4==l&& 3≠g && l≠f&& k==g && k≠f && 3==j && j==e &&i≠f;*
e) *4==l&& 3≠g && l≠f&& k==g && k≠f &&3==j &&j≠e &&i==f;*

Wir schreiben sie ein wenig um (die graumarkierten Aussagen schliessen einander aus):

a) *3==j && j==e &&3==g&& f==l&&4≠l&& g≠k && k≠f && i≠f;*
b) *3==j &&3==g && i==f && k==g && 4≠l&& l≠f&& k≠f && j≠e;*
c) *4==l&&3==j &&j==e && f==l && 3≠g &&g≠k && k≠f &&i≠f;*
d) *4==l&& 3==j && j==e && k==g&& 3≠g && l≠f && k≠f &&i≠f;*
e) *4==l&&i==f && k==g &&3==j && 3≠g && l≠f && k≠f && j≠e;*

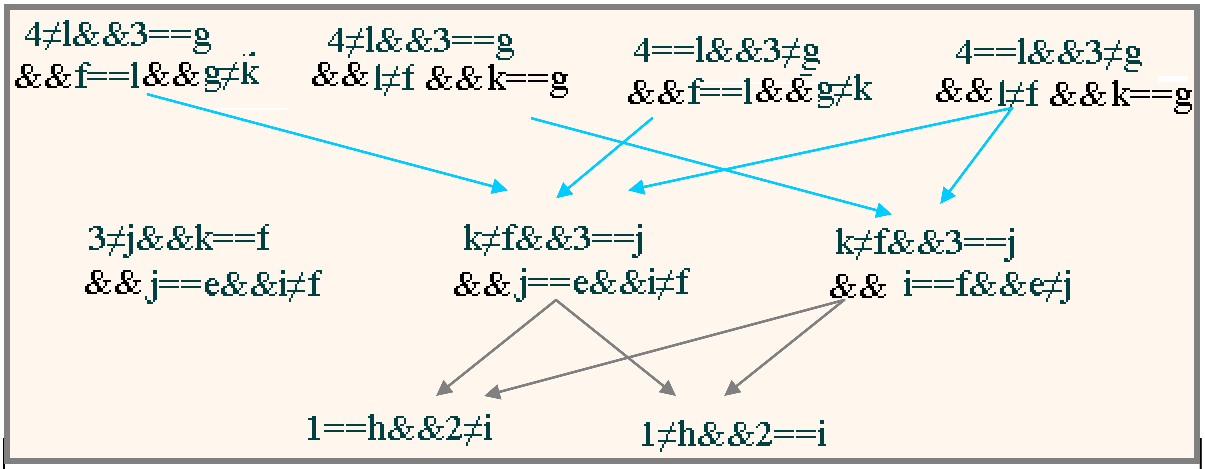


Abbildung zum Lösungsverfahren 3, Schritt 2

Die Lösungen c) und e) enthalten keine Widersprüche. Also müssen wir sie nehmen, um weiter zu kommen.

c) $4==l \&\& f==l \&\& 3==j \&\& j==e \&\& g\neq k \Rightarrow$ *g, h, i, k, 1, 2 bleiben undefiniert,* $g\neq k$;
$\Rightarrow g\neq k \Rightarrow g==i \;\&\&\; h==k \;\&\&(i==1||h==1)\&\&(i==2||h==2)$: weder (1==i&&2≠h) noch (1≠h&&2==i) erfüllen diese Bedingungen. Diese Lösung muss somit falsch sein.

e) $4==l\&\&i==f \&\& k==g \&\&3==j \&\& j\neq e \Rightarrow$ *e, f, g, h, i, k, 1, 2, 3, 4 bleiben undefiniert,* $i==f \&\& k==g \&\& j\neq e$;
$\Rightarrow 4==l\&\&3==j \&\& k==g\&\&k\neq 4\&\&k\neq 3\&\&i==f\&\&i\neq 3\&\&i\neq 4\&\&e\neq 3 \Leftrightarrow$
$4==l\&\&3==j \&\& k==g\&\&(k==1||k==2)\&\&i==f\&\&(i==1||i==2)\&\&e\neq 3 \Leftrightarrow$
$4==l\&\&3==j \&\& e==4\&\&h==3\&\& k==g\&\&(k==1||k==2)\&\&i==f\&\&(i==1||i==2)$
Ganz unten im Schema haben wir h==1&&(..) und h≠1&&(..). Für diesen Fall müssen wir die zweite Bedingung nehmen:
$4==l\&\&3==j \&\& e==4\&\&h==3\&\& k==g\&\&i==f\&\&(1\neq h\&\&2==i) \Leftrightarrow$
$l==e\&\&l==4\&\&j==h\&\&j==3\&\&i==f\&\&i==2\&\&k==g\&\&k==1$

Lösung: Amann heisst Gustav und ist mit Klara verheiratet, Bemann heisst Fritz und ist mit Inge verheiratet, Cemann heisst Heinrich und ist mit Josepha verheiratet und Demann heisst Erich und er ist mit Luise verheiratet.

4. Aufstellen von Logeleien

Ich habe nicht nur die Lösungsverfahren, sondern auch die Aufstellungsmethoden von Logeleien entwickelt und erarbeitet.
Wir wissen jetzt, wie wir Logeleien auf verschiedene Arten lösen können. Diese Kenntnisse gebrauchen wir hier, um selber solche Aufgaben aufzubauen. Wir müssen nur in der umgekehrten Reihenfolge vorgehen: anstatt von der Logelei zur Lösung umgekehrt von der Lösung zur Logelei!
Wir schauen uns auch an, wie wir eher schwierige Logeleien aufstellen können.

Das Programm zum Lösen meiner Logeleien ist im Anhang zu finden.

4.1. Aufbau von Logeleien ohne Operation „Oder“

Am Anfang schauen wir uns an, wie die Logeleien ohne „Oder“ funktionieren. Sie enthalten meistens n Gegenstände oder Personen, die n*k Eigenschaften haben. Ein einfaches Beispiel wären dafür 3 Personen mit 3*2=6 Eigenschaften. Wir zeichnen eine Tabelle:

Tabelle für 3 Personen (1, 2, 3) und ihre 6 Eigenschaften (a bis f)

	a	b	c	d	e	f
1	o	x	x	x	o	x
2	x	x	o	o	x	x
3	x	o	x	x	x	o

Wir wissen, dass jeder Zahl zwei Variablen zugeordnet werden, und zwar eine von a bis c und eine von d bis f. Wir definieren die Variablen und Werte folgendermassen: a=1, b=3, c=2, d=2, e=1, f=3. Wir schreiben es ein wenig um:
a==1&& b==3&& c==2&& d==c&& e==a&& f==b.
Wenn wir a und b kennen, dann kennen wir auch automatisch c; wir brauchen also nicht, die dritte Anordnung aus drei möglichen aufzuschreiben ⇔ *a==1&&b==3&&d==c&&e==a.*
Wenn wir a definiert haben, bleiben uns zwei möglichen Anordnungen für b. Schliessen wir die falsche aus, so bleibt b immer noch genau definiert. Mit d, e und f ist es genau gleich ⇔
a==1&&b≠2&&e==a&&d≠b

Jetzt stellen wir eine anspruchsvollere Aufgabe auf. Wir nehmen 4 Werte und 12 Variablen. Jedem Wert werden je drei Variablen zugeordnet, und zwar eine von a bis d, eine von e bis h und eine von i bis l. Zuerst zeichnen wir die Tabelle und definieren unsere Werte:

Tabelle für 4 Werte (Personen oder Gegenstände) und 12 Variablen (ihren Eigenschaften)

	a	b	c	d	e	f	g	h	i	j	k	l
1	x	o	x	x	x	x	o	x	x	x	x	o
2	o	x	x	x	x	o	x	x	x	x	o	x
3	x	x	x	o	o	x	x	x	o	x	x	x
4	x	x	o	x	x	x	x	o	x	o	x	x

*a==2&&b==1&&c==4&&**d==3**&&e==3&&f==2&&g==1&&**h==4**&&i==3&&j==4&
& k==2&&**l==1*** ⇔
*a==2&&b==1&&**c==4**&&e==3&&f==2&&**g==1**&&i==3&&j==4&&**k==2*** ⇔
*a==2&&b==1&&**c≠3**&&e==3&&f==2&&**g≠4**&&i==3&&j==4&&**k≠1**.*

Jetzt können wir im zweiten Teil (Variablen e bis h) die angeordneten Werte durch die entsprechenden Variablen aus dem ersten Teil (Variablen a bis d) ersetzen. Denn wir haben bereits alles im ersten Teil genau definiert. Genau das Gleiche machen wir mit dem dritten Teil: wir können hier Werte durch Variablen aus dem ersten Teil, oder aber auch aus dem zweiten nehmen.

⇔ *a==2&&b==1&&c≠3&&e==**d**&&f==**a**&&g≠**j**&&i==**e**&&j==4&&k≠**g***

Wir schreiben die gleichen Variablen nacheinander und vertauschen ihre Werte so, dass sie ihren Sinn nicht verlieren, aber anders aussehen:

⇔ ***(a==2&&f==a)*** *&&b==1&&c≠3&&e==d &&i==e &&g≠j &&j==4&&k≠g*

⇔ ***(a==f&&a==2&&f==2)*** *&&b==1&&c≠3&&e==d &&i==e &&g≠j &&j==4&&k≠g*

⇔ ***(a==f&&f==2)*** *&&b==1&&c≠3&&e==d &&i==e &&g≠j &&j==4&&k≠g*

Am Schluss ändern wir die Reihenfolge so, dass die Variablen nicht in der alphabetischen Reihenfolge vorkommen:

> ***⇔ i==e&& a==f&&k≠g &&c≠3&&g≠j &&e==d && f==2 &&j==4&& b==1***

Ersetzen wir Zahlen durch Personen und Variablen durch ihre Eigenschaften, bekommen wir eine Logelei:
1 sei Frau Lustig, 2 sei Herr Triste, 3 sei Herr Alleswisser, 4 sei Frau Modern;
a sei rothaarig, b sei blond, c sei schwarzhaarig, d sei grünhaarig;
e sei Schreibtisch, f sei Sofa, g sei Esstisch, h sei Fernseher;
i sei Igel, j sei Affe, k sei Papagei, l sei Meerschweinchen.

Unsere Logelei 1:

„Frau Lustig, Herr Triste, Herr Alleswisser und Frau Modern sind benachbart. Aber sie sind so verschieden, dass es einen Mann Wunder genommen hat, wie jeder von ihnen heißt und worin sie sich unterscheiden. Eine Frau, die alle diese Personen gut kannte, hat sich engagiert, ihm zu helfen, in der Hoffnung, dass er nicht gleich alles herausfindet : „Es ist bekannt, dass eine der Personen rothaarig, eine blond, eine Schwarzhaarig und eine grünhaarig ist. In der Mitte ihrer Wohnzimmer steht je ein Gegenstand: ein Schreibtisch, ein Sofa, ein Esstisch oder ein Fernseher. Sie besitzen je ein Tier: einen Igel, eine Affe, einen Papagei oder ein Meersschweinchen.
Die Person, die den Igel zu Hause hat, hat einen Schreibtisch in der Mitte des Wohnzimmers.
`i=e`
Bei der rothaarigen Person steht das Sofa in der Mitte des Wohnzimmers.
`a==f`
Diejenige Person, die den Papagei besitzt, hat keinen Esstisch im Wohnzimmer.
`k≠g`
Herr Alleswisser hat kein schwarzes Haar.
`c≠3`
Diejenige Person, die den Esstisch im Wohnzimmer hat, mag keine Affen.
`g≠j`
Diejenige Person, welche den Schreibtisch im Wohnzimmer gestellt hat, ist grünhaarig.
`e==d`
Herr Triste hat ein Sofa in der Mitte des Wohnzimmers.
`f==2`
Frau Modern hat die Affe.
`j==4`
Frau Lustig ist blond.“
`b==1`
Nun hat sich diese Frau geirrt: der Mann weiß jetzt nach all diesen Angaben ganz genau welche der Personen was für eine Haarfarbe hat, welches Tier besitzt und was für einen Gegenstand im Wohnzimmer hat. Wissen Sie es auch?“

4.2. Aufbau von Logeleien basiert auf Lösungsverfahren 1

Wenn wir diverse Bedingungen in einer Aufgabe haben, verbinden wir sie durch „&&". Somit haben wir am Anfang A&&B&&C&&(..). Wir wissen jetzt, dass eine Logelei mit „Oder" schlussendlich in die folgende Form gebracht werden kann: D||E||F||(..). Jede der Aussagen ist dabei eine mögliche Lösung der ganzen Aufgabe.
Wir fangen mit der Lösung an, und bauen schrittweise die Logeleien auf, die nur eine richtige Lösung haben.

4.2.1. Logelei der Ursprungsform (A||B)&&C&&D

Wir fangen mit einer einfachen Aufgabe an: (A||B)&&C&&D. Diese Aufgabe hat 2 Lösungen: (A&&C&&D) || (B&&C&&D). Die erste Lösung sei richtig und die zweite falsch. Der einzige Unterschied zwischen den beiden liegt in den Aussagen A und B. A sollte also mit (C **und** D) eine Lösung bilden, und B mit (C **und** D) keine.

4.2.1.1. Die Anzahl der Zahlen gleich der Anzahl der Variablen

Nehmen wir 4 Variablen und 4 Zahlen: a, b, c, d und 1, 2, 3, 4. a=1, b=2, c=3 und d=4 sei die Lösung. Dann: A sei a=1, C sei b=2 und D sei c==3&&d==4. Wir haben also 2 Lösungen: (a==1&&b==2&&c==3&&d==4)||(B&&b==2&&c==3&&d==4). Wie soll jetzt B gewählt werden? Wir wissen über die zweite Lösung, dass sie falsch sein muss, also, muss B (b==2&&c==3&&d==4) ausschließen. B könnte z.B. b≠2 oder d==3 sein, weil es unserer eindeutigen Anordnung widersprechen würde.
Wir wollen aber, dass der Leser unsere Aufgabe nicht gleich löst, sondern, dass er möglichst lange dabei überlegt. Darum schreiben wir die gleiche Lösung anders auf: a=1 bedeutet a≠2, a≠3, a≠4, b=2 bedeutet b≠1, b≠3, b≠4, c=3 bedeutet c≠1, c≠2, c≠4 und d=4 bedeutet d≠1, d≠2, d≠3. Also, können wir schreiben:
a≠2& a≠3&&a≠4&& b≠1&&b≠3&&b≠4&& c≠1&&c≠2&&c≠4&& d≠1&&d≠2&&d≠3
Für a braucht es 3 Bedingungen, für b aber 3-1= 2 Bedingungen, weil wenn wir a=1 wissen, wissen wir bereits, dass b≠1 ist, c braucht 2-1= eine Bedingung, weil c≠1 aus a=1 und c≠2 aus b=2 folgen. d braucht dagegen 1-1= keine Bedingungen. Also:
a≠2 && a≠3 && a≠4 && b≠3 && b≠4 && c≠4
Jetzt vertauschen wir die Reihenfolge, damit es nicht zu auffällig erscheint:
a≠2 && c≠4 && 4≠b && a≠3 && a≠4 &&3≠b
A sei jetzt a≠2, C sei (*c≠4 && 4≠b && a≠3)* und *D* sei (*a≠4 &&3≠b).* Dann haben wir:
B && *c≠4 && 4≠b && a≠3 && a≠4 &&3≠b*
Wir zeichnen eine Tabelle mit den Werten und Variablen für C&&D (S.40).
Aus der Tabelle sehen wir, dass die unmöglichen Lösungen sind: c==1, c==2, d==1, d==2, d==3 (weil a und b unbedingt 1 oder 2 sein müssen, und somit c=3) und alle Anordnungen, die bereits als falsch angekreuzt sind. So können wir für B eine dieser falschen Anordnungen nehmen. Wir nehmen c==1. Denn c≠1 kommt weder in C noch in D vor. Somit lautet unsere Aufgabe folgendermaßen:

(a≠2||c==1) && (c≠4 && 4≠b && a≠3) && (a≠4 &&3≠b)

Jetzt können wir sie noch umformen:

⇔ *(a≠2||c==1) && (c≠4 && (4≠b&&3≠b) && (a≠3 && a≠4))*
⇔ *(a≠2||c==1)&& c≠4 && (b==1||b==2) && (a==1||a==2)*
⇔ *(a≠2||c==1)&& c≠4 && (b==1&&a==2)|| (a==1&&b==2)*

Tabelle für C&&D (c≠4 && 4≠b && a≠3 && a≠4 &&3≠b)

	a	b	c	d
1				
2				
3	x	x		
4	x	x	x	

> ***⇔ Implies[a==2, c==1] && c≠4 && Implies[b≠1||a≠2, a==1&&b==2]***

Jetzt können wir Variablen durch Personen und Zahlen durch Haustiere ersetzen:
a sei Frau Geisthaber, b sei Herr Langweiler, c sei Frau Unmöglich und d sei Herr Ruhemann; 1 sei die Eule, 2 sei der Bär, 3 sei Eichhörnchen und 4 sei der Flusspferd.

Unsere Logelei 2:

„Frau Geisthaber, Herr Langweiler, Frau Unmöglich und Herr Ruhemann haben ihre Tiere für einen Tag ins Tierheim abgegeben. Jedoch haben sie vergessen, dem Tierheimpräsident mitzuteilen, welches Tier wem gehört.
Er hat aber gehört, dass wenn Frau Geisthaber einen Bär besitzt, dann hat Frau Unmöglich eine Eule in ihrem Wohnzimmer.
`Implies[a==2, c==1]`
Es ist allen bekannt, dass Frau Unmöglich keine Flusspferde mag.
`c≠4`
Wenn Herr Langweiler keine Eule besitzt oder wenn Frau Geisthaber keinen Bär hat, dann haben Frau Geisthaber eine Eule und Herr Langweiler einen Bär.
`Implies[b≠1||a≠2, a==1&&b==2]`
Nach einiger Zeit Überlegung wusste der Präsident, wem welches Tier gehört. Wissen Sie es auch?“

4.2.1.2. Die Anzahl der Zahlen ungleich der Anzahl der Variablen

Nehmen wir 4 Zahlen und 8 Variablen: von 1 bis 4 und a bis h. Es gelte: 1=b, 1=g, 2=a, 2=h, 3=d, 3=e, 4=c und 4=f. Wir zeichnen eine Tabelle dazu:

Tabelle für *1=b, 1=g, 2=a, 2=h, 3=d, 3=e, 4=c und 4=f*

	a	b	c	d	e	f	g	h
1	x	o	x	x	x	x	o	x
2	o	x	x	x	x	x	x	o
3	x	x	x	o	o	x	x	x
4	x	x	o	x	x	o	x	x

Wir haben: *2==a&&1==b &&4==c &&3==d &&2==h &&3==e &&1==g &&4==f.*
Wir möchten, dass unsere Logelei schwierig ist, darum schreiben wir diese Anordnung ein wenig anders auf. Wir haben bereits in den Kapiteln 4.1. und 4.2.1.1 angeschaut, wie wir solche Bedingungen anders aufschreiben können :
⇔ *a≠1&& a≠3&& a≠4&& b≠3&& b≠4&& c≠3 && 2==h&& 3==e&& 1==g && 4==f*
⇔ *a≠1&& a≠3&& a≠4&& b≠3&& b≠4&& c≠3 && a==h&& d==e && b==g&& c==f*
⇔ *a≠1&& a≠3&& a≠4&& b≠3&& b≠4&& c≠3 && a==h&& d==e&& b==g*
⇔ *a≠1&& a≠3&& a≠4 && b≠3&& b≠4&& c≠3&& a==h && d==e && b≠f*

Wir vertauschen einige Variablen und ihre Werte so, dass sie ihren Sinn nicht verlieren:

⇔ *(a≠1&& a==h && h≠3&& h≠4) && d==e && (b≠3&& b≠f && g≠4) &&(f≠3)*

Wir verteilen unsere Anordnungen in Aussagen:
A sei (*a≠1&& h≠3), C* sei (*h≠4&&b≠3&& g≠4&& b≠f)* und D sei *(f≠3&& d==e&&a==h).* Dann haben wir: *((a≠1&&h≠3) || B) && (b≠3&& g≠4) && (f≠3&& d==e&&a==h).* Die falsche Lösung ist B && (*b≠3&& g≠4*) && (*f≠3&& d==e&&a==h*).
Wie wählen wir am einfachsten B aus, so dass (C **und** D) nicht stimmen, aber auch so, dass es nicht auffällt, dass B zu (C **und** D) nicht passt? Wir schreiben zwei Bedingungen auf, die wir für unsere Lösung haben. Sie sind genau gleich, nur verschieden aufgeschrieben: Die erste Bedingung haben wir am Schluss und die zweite am Anfang aufgeschrieben:

- *a≠1&& h≠3&& h≠4&& a==h&& b≠3&& g≠4&& b≠f&& f≠3&& d==e*
- *a≠1&&a≠3&&a≠4&&b≠3&&**b≠4&&c≠3** && **b==g**&&a==h&&d==e&&**c==f***

Den beiden Bedingungen nehmen wir jetzt die ersten 2 Anordnungen weg: a≠1&&a≠3 bzw. a≠1&&h≠3. (Wir können a durch h ersetzen, weil die Anordnung a==h in beiden Bedingungen vorkommt und unverändert bleibt.)

(Achtung: Wir dürfen für diesen Fall **nicht** die Bedingungen nehmen, in denen der erste Teil von a bis d nicht genau gleich ist.
Wenn wir eine Anordnung entnehmen wollen, die 2 Variablen enthält wie z.B. *b==g*, dann kann der erste Teil von a bis d verschieden aufgeschrieben sein, aber nicht der zweite Teil, der b==g enthält, er muss unverändert bleiben. Wir können also dann die folgenden Bedingungen nehmen und gleichsetzen:

- *a≠1&&a≠3&&a≠4&&b≠3&&b≠4&&c≠3 && b==g&&a==h&&d==e&&c==f*
- *a≠1&&a≠3&&a≠4&&b≠2&&b≠3&&b≠4&&c≠1&&c≠2&&c≠3&&d≠1&&d≠2&&d ≠4&& b==g&&a==h&&d==e&&c==f*

Wenn wir etwas entnehmen wollen, das beide Arten von Anordnungen enthält wie z.B. a≠1&&b==g, dann können wir keine Bedingungen mehr gleichsetzen, denn sie würden jetzt völlig andere möglichen Lösungen haben. Darum ist dieser Fall ungünstig.)

Wir vergleichen die erste Bedingung mit der zweiten: b==g, b≠4, c==f, c≠3 kommen in der ersten nicht vor. Also können wir für B das Gegenteil einer diesen Anordnungen nehmen, damit (B und C und D) zur falschen Aussage wird.
Wir können für B (c==3) nehmen, oder (b==4), oder alles zusammen: (b≠g||b==4||c≠f||c==3). Der Leser hätte dann zwar viel zum Überlegen, aber die Logelei würde komisch tönen: „.. oder... oder.. oder...oder..“.
Wir nehmen (b==4||c≠f) für B.

((a≠1&& h≠3)||(b==4||c≠f) && h≠4&& a==h&& b≠3&& g≠4&& b≠f&& f≠3&& d==e
⇔ *Implies[a==1||h==3,b==4||c≠f]&& (g≠4&& h≠4)&& (a==h)&& (d==e)&& (b≠3&&b≠f)&& (f≠3)*

> → ***Implies[a==1||h==3,b==4||c≠f]&& Xor[4==e,4==f]&& (a==h)&& (d==e)&& (b≠3&&b≠f)&& (f≠3)***

Wir nehmen 4 Männer, 4 Badehosenfarben und 4 Getränke.
1 sei der Tennisspieler, 2 sei der Schwimmer, 3 sei der Radfahrer, 4 sei der Fußballspieler;
a sei grün, b sei blau, c sei rot und d sei gelb;
e sei Bier, f sei Coca-Cola, g sei Mineralwasser und h sei Orangensaft.

Unsere Logelei 3:

„In einem Hotel im Süden des Frankreichs wurden 4 Sportler angemeldet: ein Tennisspieler, ein Schwimmer, ein Radfahrer und ein Fußballspieler. Es wurde bekannt, dass heute alle vier am Strand zu sehen sind. Leider haben sie alle schwarze Brillen an, weil sie nicht erkennt werden möchten. Ihre Fans wollen aber trotzdem ein Autogramm. Darum haben sie folgende Informationen gesammelt, um ihre Idole zu erkennen:
„Wenn der Mann in der grünen Badehose der Tennisspieler ist oder wenn der Mann mit dem Orangensaft in der Hand der Radfahrer ist, dann ist der Mann in der blauen Badehose der Fußballspieler oder dann trinkt der Mann in der roten Badehose kein Coca-Cola.
`Implies[a==1||h==3,b==4||c≠f]`
Der Fußballspieler trinkt entweder Bier oder Coca-Cola.
`Xor[ 4==e,4==f]`
Der Mann in der grünen Badehose hat ein Glas Orangensaft in der Hand.
`a==h`
Der Mann in der gelben Badehose ist am Biertrinken.
`d==e`
Der Radfahrer mag keine blauen Badehosen und der Mann in der blauen Badehose mag kein Coca-Cola.
`b≠3&&b≠f`
Der Radfahrer mag auch kein Coca-Cola.“
`f≠3`
Helfen Sie den Fans zu erkennen, was jeder der Sportler trinkt und was für eine Farbe seine Badehose hat?“

4.2.2. Logelei der Ursprungsform (A||B)&&(C||D)&&E

Wir bauen eine schwierigere Logelei auf, welche in ihrer Ursprungsform zwei „||“ zwischen „&&“ hat. Dies bedeutet, dass sie 4 mögliche Lösungen hat.
Die Ursprungsform ist (A||B)&&(C||D)&&E
Wir nehmen an, zu unserer richtigen Lösung gehören die ersten Aussagen der Bedingungen in den Klammern, dann sollen die zweiten Aussagen so gewählt werden, dass sie Lösungen bilden, die falsch oder ausgeschlossen sind.
(A||B)&&(C||D)&&E ⇔ *(A&&C&&E)|| (A&&D&&E)|| (B&&C&&E)|| (B&&D&&E)*.
(A&&C&&E) soll bei uns die einzig richtige Lösung sein. (A&&D&&E) ist dann eine der möglichen Lösungen, die falsch sein muss. Damit diese Lösung ausgeschlossen wird, muss D A oder E widersprechen. D kommt aber noch einmal vor und zwar in (B&&D&&E). Damit diese Möglichkeit ausgeschlossen werden kann, muss D auch B oder E widersprechen. Also muss D die Bedingung ((A&&E)||(B&&E)) ausschließen . Mit B sieht es ähnlich aus: B muss (C&&E) **und** (D&&E) ausschließen.

Wir nehmen 3 Zahlen und 9 Variablen und ordnen sie folgendermaßen an:

Tabelle für 1=b=f=i, 2=a=d=g und 3=c=e=h

	a	b	c	d	e	f	g	h	i
1	x	o	x	x	x	o	x	x	o
2	o	x	x	o	x	x	o	x	x
3	x	x	o	x	o	x	x	o	x

$a==2\&\&b\neq2\&\&b\neq3\&\&c\neq1\&\&c\neq2\&\&d==a\&\&e\neq b\&\&e\neq a\&\&f==b\&\&h==c\&\&i\neq d$

Wir brauchen es nicht, alle Bedingungen aufzuschreiben:

$\Leftrightarrow a==2\&\&b\neq3\&\&d==a\&\&e\neq b\&\&h==c\&\&i\neq d$

Für A nehmen wir $(h==c)$, für C $(a==2\&\&b\neq)3$, und für E den Rest: $(d==a\&\&e\neq b\&\&i\neq d)$. Jetzt müssen wir B auswählen. B soll mit (D&&E) und mit (C&&E) keine Lösung bilden. D ist noch unbekannt, darum schreiben wir zuerst C&&E auf: $(a==2\&\&\ b\neq3\&\&\ d==a\&\&\ e\neq b\&\&\ i\neq d)$.
Wir wollen die umgeformte Bedingung (C&&E) ihrer ursprünglichen Form gleichsetzen, darum nehmen wir die ganze Bedingung unumgeformt und umgeformt so, dass wir ihr die Aussage A (h==c) wegnehmen können. Somit muss der dritter Teil, der „h“ enthält, in der ursprünglichen Form und umgeformt gleich bleiben. Der erste Teil kann dagegen verschieden sein:

- $a==2\&\&\mathbf{b\neq2}\&\&b\neq3\&\&\mathbf{c\neq1}\&\&\mathbf{c\neq2}\&\&d==a\&\&e\neq b\&\&\mathbf{e\neq a}\&\&\mathbf{f==b}\ \&\&h==c\&\&i\neq d$
- $a==2\&\&\ b\neq3\&\&\ d==a\&\&\ e\neq b\&\&h==c\&\&\ i\neq d$

B soll das Gegenteil einer der Aussagen aus der ersten Bedingung sein, die nicht in der zweiten vorkommt. Z.B. (c==1) oder (e==a). Wir nehmen das erste: c==1.
Für D gilt: D soll mit (B&&E) und mit (A&&E) unmögliche Lösungen bilden. (B&&E) ist $(c==1\&\&\ d==a\&\&e\neq b\&\&i\neq d)$, (A&&E) ist $(h==c\ \&\&\ d==a\&\&e\neq b\&\&i\neq d)$. Das Gemeinsame, was sie haben ist E. Am einfachsten wäre es, wenn D die Aussage E ausschließen würde. Dies ist aber sehr auffällig, darum nehmen wir für D das Gegenteil von (A||B): (1≠c&&h≠c) oder als Alternative (2==c&&h==1). Wir nehmen die zweite Aussage.

$(h==c||\ c==1)\ \&\&\ ((a==2\&\&b\neq3)||\ (2==c\&\&h==1))\ \&\&\ d==a\&\&e\neq b\&\&i\neq d$

da a==d genau definiert ist, können wir a durch d ersetzen:

$\Leftrightarrow (h==c||\ c==1)\ \&\&\ ((d==2\&\&b\neq3)||\ (2==c\&\&h==1))\ \&\&\ d==a\&\&e\neq b\&\&i\neq d$

> $\rightarrow \mathbf{Implies[h\neq c,c==1]\ \&\&\ Implies[d\neq2||b==3,\ 2==c\&\&h==1]\ \&\&d==a\&\&e\neq b\&\&i\neq d}$

1 sei Herr Gärtner, 2 sei Herr Ingenieur und 3 sei Herr Baumann;
a sei die Katze, b sei der Hund und c sei der Papagei;
d sei das Auto, e sei das Velo und f sei der Motorrad;
g sei das Kaffe, h sei der Tee und i sei der Orangensaft.

Unsere Logelei 4:

„Drei Herren wohnen nebeneinander: Herr Gärtner, Herr Ingenieur und Herr Baumann. Sie sind ganz verschieden und man weiß über sie, dass sie je ein Tier haben, immer dasselbe Transportmittel gebrauchen und immer nur ein Getränk am morgen trinken.
Der Mann, der mit dem Auto zur Arbeit geht, hat eine Katze.
`d==a`
Der Mann, der am morgen Orangensaft trinkt, fährt nicht mit dem Auto.
`i≠d`
Der Mann, der mit dem Velo fährt, hat keinen Hund.
`e≠b`
Wenn der Mann mit dem Auto nicht Herr Ingenieur heißt, oder wenn Herr Baumann einen Hund hat, dann hat Herr Ingenieur einen Papagei und Herr Gärtner trinkt Kaffe am morgen.
`Implies[d≠2||b==3, 2==c&&h==1]`

Wenn der Mann, der am morgen Tee trinkt, keinen Papagei hat, dann hat Herr Gärtner den Papagei.
`Implies[h≠c,c==1]`
Wer trinkt was am morgen, welches Transportmittel gebraucht er und welches Tier besitzt er? "

4.2.3. Logelei der Ursprungsform (A||B)&&(C||D)&&(E||F)

Wir haben (A||B)&&(C||D)&&(E||F). Damit die Logelei möglichst schwierig ist, müssen A, B, C, D, E und F möglichst verschieden sein und diese Aufgabe soll nur eine Lösung haben. Denn ist z.B. A gleich C, dann haben wir (A||B)&&(A||D)→ (A)&&(B||D): da haben wir weniger „||", was das Lösen von dieser Aufgabe einfacher macht. Wenn wir mehrere Lösungen haben, ist die Wahrscheinlichkeit größer, irgendeine Lösung schnell zu finden.
Zu unserer Lösung gehöre: A&&C&&E. Alle andere Lösungen müssen dann unmöglich sein. In diesem Fall muss B (C || D) oder (E || F) widersprechen, D (A || B) oder (E || F) und F muss (C || D) oder (A || B) widersprechen. Damit unsere Aussagen sich nicht wiederholen, müssen wir verschiedene Aussagen nehmen. Somit soll B (E||F) widersprechen, D (A||B) und F soll dann (C||D) widersprechen:

- B → Widerspruch zu E und zu F
- D → Widerspruch zu A und zu B
- F → Widerspruch zu C und zu D

Wir nehmen an, wir haben 4 Variablen und 4 Zahlen: a bis d und 1 bis 4. Jede Variable kann je einen Wert annehmen.
Die richtige Lösung sei: a=3, b=2, c=1, d=4. Jetzt können wir variieren. Wir können sie gleich so als eine Lösung aufschreiben *(a=3&&b=2&&c=1&&d=4)*, oder als
(a≠2&&a≠1&&a≠3&&b≠1&&b≠4&&c≠4) oder *(a==3&&b≠1&&d≠1&&b≠4)* usw.
Wenn wir *(a≠1&&a≠2&&a≠4&&b≠1&&b≠4&&c≠4)* haben, können wir für die Aussage A *(a≠1&&a≠2&&a≠4)* und für B *(b≠1)* nehmen, oder aber auch *(a≠2&&a≠4&&b≠1)* für A und *(a≠1)* für B. Aus dem ersten Beispiel sehen wir gleich, dass a=3 ist, aus dem zweiten aber nicht. Je weniger wir uns an die Reihenfolge halten, desto schwieriger erscheint die Aufgabe.
Wir nehmen für unsere Logelei *(a==3&&b≠1&&d≠1&&b≠4),* für A nehmen wir *a==3*, für C *b≠4*, für E *d≠1*. Die Aussage *b≠1* können wir als eine zusätzliche festgelegte Bedingung nehmen, damit unsere Aufgabe mehre Bedingungen hat und es einfacher fällt, B, D und F zu finden.
(a==3||B) && (b≠4||D) && (b≠1||F) && d≠1.
Wir sehen, dass wir viele b haben, darum sollen wir versuchen, die anderen Variablen mehr zu gebrauchen.

- D → Widerspruch zu A und zu B

Für D nehmen wir z.B. 3==d&&1==c, da wir B noch nicht kennen und es A widerspricht.

Als Nächstes nehmen wir die Aussage, welche die Aussage D ausschließt.

- F → Widerspruch zu C und zu D

Wir nehmen z.B. als Widerspruch zu D (d==1)||(c==3). Aber es kann auch eine andere Aussage sein, z.B. nur (d==1) oder (3==a&&c==2). Als Widerspruch zu C nehmen wir (b==4). Also, F ist (d==1||c==3)&&(b==4).

- B → Widerspruch zu E und zu F

$(b\neq 1||((d==1||c==3)\&\&(b==4)))$ soll widergesprochen werden. Also muss b gleich 1 sein, dann ist F $(b\neq 4)$ widergesprochen. Also müssen wir eine Bedingung aufstellen, die keine b enthält, aber dazu führt, dass b gleich 1 sein muss. Z.B. $\underline{(1\neq c\&\&1\neq d\&\&a==2)}$.

Also, unser Logelei sieht so aus:

(a==3||(1≠c&&1≠d&&a==2)) && (b≠4||(3==d&&1==c)) && (b≠1||((d==1||c==3)&&(b==4)) && d≠1

> **⇔ *Implies[1==c||1==d||a≠2, a==3] && Implies[b==4, 3==d&&1==c] && Implies[(d≠1&&c≠3)||b≠4, b≠1] && d≠1***

Jetzt schreiben wir unsere Logelei in der deutschen Sprache auf.

1 bis 4 seien 4 Kosmonauten, a bis d seien Geschenke, die sie für ihre Frauen aus dem Weltall mitgebracht haben.
Kosmonaut Armstrong sei 1, Kosmonaut Gagarin sei 2, Kosmonaut Zukunftsglocke sei 3 und Kosmonaut Meyer sei 4.
a sei Gold des Plutoniums, b sei ein Foto der Erde, c sei ein Mondstückchen, d sei eine Venusblume.

Unsere Logelei 5:

„Die Kosmonauten Armstrong, Gagarin, Zukunftsglocke und Meyer sind aus dem Weltall zurückgekommen. Ihre Frauen freuen sich am meisten, weil sie auch Geschenke erhalten. Man weiss, dass alle Kosmonauten an verschiedenen Orten waren und je ein Geschenk von diesem Ort mitgebracht haben.
Die Medien verschweigen jedoch, wer wo war und beziehungsweise was mitgebracht hat.
Es wurde bekannt, dass wenn Armstrong entweder ein Mondstückchen oder eine Venusblume mitgebracht hat oder wenn Gagarin kein Foto der Erde gemacht hat, dann hat Zukunftsglocke Gold des Plutoniums mitgebracht.
`(Implies[1==c||1==d||a≠2, a==3])`
Wenn Meyer ein Foto der Erde gemacht hat, dann war Zukunftsglocke auf der Venus und Armstrong auf dem Mond.
`(Implies[b==4, 3==d&&1==c])`
Wenn Armstrong keine Venusblume und Zukunftsglocke kein Mondtückchen mitgebracht haben oder wenn Meyer kein Foto der Erde mitgebracht hat, dann hat Armstrong auch kein Foto der Erde gemacht.
`(Implies[(d≠1&&c≠3)||b≠4, b≠1])`
Armstrong hat im letzten Interview gesagt, dass er Venus nicht gesehen hat.
(d≠1)
Nun denken Journalisten, dass das Publikum immer noch nicht weiss wer was mitgebracht hat. Doch sie unterschätzen das begeisterte Volk. Wer hat was mitgebracht?“

4.3. Aufbau von Logeleien basiert auf Lösungsverfahren 2

In dieser Methode analogisch wie im Lösungsverfahren 2 schreiben wir alle möglichen Lösungen in die Tabelle auf, um auf die Lösung schrittweise zu kommen. Jetzt versuchen wir, von so einer Tabelle auszugehen, um Logeleien selber aufzubauen.

4.3.1. Logelei mit 4 Zahlen und 4 Variablen

Wir nehmen a,b,c, d und 1,2,3, 4. Die Anzahl der Möglichkeiten ist gleich 4*3*2*1 = 24.
Wir zeichnen eine Tabelle, die alle Möglichkeiten enthält, und lassen den Platz frei für die Bedingungen:

Tabelle zum Aufbau von Logeleien mit 4 Zahlen und 4 Variablen

a	b	c	d		b1	b2	b3	b4
1	2	3	4		x			
1	2	4	3					
1	3	2	4					
1	3	4	2					
1	4	2	3					
1	4	3	2		x			
2	1	3	4		x	x	x	
2	1	4	3		x	x		
2	3	1	4		x			
2	3	4	1		x	x		
2	4	1	3		x			
2	4	3	1		x			
3	1	2	4					
3	1	4	2					
3	2	1	4					
3	2	4	1					
3	4	1	2					
3	4	2	1					
4	1	2	3					
4	1	3	2		x	x	x	x
4	2	1	3					
4	2	3	1		x			
4	3	1	2					
4	3	2	1					

Am Anfang nehmen wir irgendeine Bedingung b1, z.B. (*a==2||c==3).*
Für b2 nehmen wir beispielsweise (*b==1||c==4)*. Wir nehmen möglichst die Variablen und Zahlen, welche in den vorherigen Bedingungen nicht vorgekommen sind. Denn wenn wir *a==1* nehmen würden, hätten wir nur wenige mögliche Lösungen und zwar nur bei c==3 aus der ersten Bedingung; wenn wir dagegen *a==3* nehmen würden, hätten wir gar keine Lösungen mehr.
Für b3 schlagen wir also (*d==4||c==3)* vor.
Achtung: es können nicht irgendwelche Anordnungen sein. Sie müssen nach der vorherigen Bedingung in den möglichen Lösungen vorkommen.

Wir haben nach der dritten Bedingung zwei Lösungen: 2,1,3,4 und 4,1,3,2.

Der Unterschied zwischen ihnen besteht in a und d. Dann können wir für die richtige Lösung (d==2) nehmen. Wir machen aber aus der b4 wieder eine Bedingung der Form (A||B) und nehmen für die zweite Aussage einfach eine beliebige Anordnung, die nach der b3 nicht mehr vorgekommen ist und die harmlos erscheint.
b4 sei somit *(d==2||b==3)*.

So haben wir die folgende Logelei bekommen:

(a==2||c==3) && (b==1||c==4) && (d==4||c==3) && (d==2||b==3)

> ***⇔ Implies[a≠2,c==3] && Implies[b≠1,c==4] && Implies[d≠4,c==3] &&Implies[d≠2,b==3]***

Nehmen wir an, a sei das Buch „Die letzte Welt", b sei das Buch „Die Angst des Tormanns beim Elfmeter", c sei „Die dunkle Seite des Mondes" und d sei „Micky Mouse".
1 sei blau, 2 sei grün, 3 sei schwarz und 4 sei rot.

Unsere Logelei 6:

> *„Auf dem Regal der Dorfbibliothek sind die letzten vier Bücher zu sehen. Man weiß nicht genau, welches Buch was beinhaltet, weil diese Bücher so oft vom Staub geputzt worden sind, dass es keine Namen mehr zu sehen sind. Jedes der vier Bücher hat nur eine Farbe und keines der Bücher hat dieselbe Farbe, wie das andere.*
> *Wenn das Buch „Die letzte Welt" nicht grün ist, dann ist das Buch „Die dunkle Seite des Mondes" schwarz.*
> `Implies[a≠2, c==3]`
> *Wenn das Buch „Die Angst des Tormanns beim Elfmeter" nicht blau ist, dann ist das Buch „Die dunkle Seite des Mondes" rot.*
> `Implies[b≠1, c==4]`
> *Wenn das Buch „Micky Mouse" nicht rot ist, dann ist das Buch „Die dunkle Seite des Mondes" schwarz.*
> `Implies[d≠4, c==3]`
> *Wenn das Buch „Micky Mouse" nicht grün ist, dann ist das Buch „Die Angst des Tormanns beim Elfmeter" schwarz.*
> `Implies[d≠2, b==3]`
> *Welche Farbe hat jedes der vier Büchern ?"*

4.3.2. Logelei mit 2 Zahlen und 5 Variablen

Wenn wir 5 Variablen und 2 Zahlen haben, können wir auch eine Tabelle zeichnen, weil sie nicht allzu gross ist (2^5= 32 Möglichkeiten).
Wir lassen wieder den freien Platz für die Bedingungen rechts.
Wir dürfen variieren vor allem bei den ersten Bedingungen.
b1 sei (*a==2||c==2)*, b2 sei (*d==2||e==2)*, b3 sei *(b==1&&c==1)||(d==1)*,
b4 sei *(d==2&&e==2)||(b==2)*, b5 sei (*b==1)*.

(a==2||c==2)&& (d==2||e==2)&& ((b==1&&c==1)||(d==1))&& ((d==2&&e==2)||(b==2))&& (b==1) ⇔

> ***⇔ Implies[a≠2,c==2] && Implies[d≠2,e==2]&&Implies[b≠1||c≠1,d==1] && Implies[d≠2||e≠2,b==2] && (b==1)***

Tabelle zur Aufgabe 1 aus dem Buch „99 Logeleien von Zweistein“

a	b	c	d	e		b1	b2	b3	b4	b5
1	1	1	1	1						
1	1	1	1	2						
1	1	1	2	1						
1	1	1	2	2						
1	1	2	1	1		x				
1	1	2	1	2		x	x	x		
1	1	2	2	1		x				
1	1	2	2	2		x	x			
1	2	1	1	1						
1	2	1	1	2						
1	2	1	2	1						
1	2	1	2	2						
1	2	2	1	1		x				
1	2	2	1	2		x	x	x	x	
1	2	2	2	1		x	x			
1	2	2	2	2		x	x			
2	1	1	1	1		x				
2	1	1	1	2		x	x	x		
2	1	1	2	1		x	x	x		
2	1	1	2	2		x	x	x	x	x
2	1	2	1	1		x				
2	1	2	1	2		x	x	x		
2	1	2	2	1		x	x			
2	1	2	2	2		x	x			
2	2	1	1	1		x				
2	2	1	1	2		x	x	x	x	
2	2	1	2	1		x	x			
2	2	1	2	2		x	x			
2	2	2	1	1		x				
2	2	2	1	2		x	x	x	x	
2	2	2	2	1		x	x			
2	2	2	2	2		x	x			

1 sei „modern“ und 2 sei „unmodern“;
a seien roten Socken, b sei rotes Haar, c sei rote Nase, d sei rotes Natel und e sei rote Hose.

Unsere Logelei 7:

„Heute haben sich die bekannten ModeratorInnen des Landes zusammengesetzt, um die neuen Trends des Jahres zu besprechen. Um sie herauszufinden sind sie von den folgenden Zusammenhängen ausgegangen:
Wenn die roten Socken nicht IN sind, dann ist die rote Nase modern.
Wenn das rote Natel unmodern sein soll, dann soll die rote Hose IN sein.
Wenn rotes Haar oder rote Nase nicht IN sind, dann ist das rote Natel IN.
Wenn das rote Natel oder rote Hose nicht OUT sind, dann muss rotes Haar OUT sein.
Rotes Haar bleibt IN.
Wissen Sie schon was die Trends von morgen sind?“

4.4. Aufbau von Logeleien basiert auf Lösungsverfahren 4

Zuerst machen wir ein einfaches Beispiel. Wir nehmen 3 Zahlen und 3 Variablen. Wir zeichnen eine Tabelle dazu, wie im Lösungsverfahren 4. Nun setzen wir zuerst die Anordnungen ein, welche zur richtigen Lösungen gehören sollen (grün): b1, b2 und b3.

	a	b	c
1	b1	b1	
2	b2	b3	b3
3		b2	

Jetzt nehmen wir eine b1, die b2 oder b3 ausschließt, damit wir keine zusätzlichen Lösungen bekommen. Z.B. bei b==1.
Die falsche b2 platzieren wir beispielsweise bei a==2, damit sie die einzige b3, die wir bis jetzt haben, ausschließt.
Für die zweite b3 nehmen wir eine Anordnung, die entweder beide b1 oder beide b2 ausschließt. Z.B. b==2.
Somit haben wir:
(a==1||b==1)&&(a==2||b==3)&&(b==2||c==2)

Machen wir ein schwierigeres Beispiel. Wir nehmen 4 Zahlen und 8 Variablen. Zuerst stellen wir eine Tabelle mit den „richtigen" Bedingungen auf (grün):

	a	b	c	d	e	f	g	h
1			b5a, b6	b4a	b4a			
2		b2		b1	b3		b5b	
3	b1					b6		b2
4	b4	b1	b3		b5a		b3	b5b

Als Nächstes suchen wir nach den falschen Bedingungen (rot).

Wir nehmen beispielsweise die alternativen falschen Bedingungen für b1 bei b==4 und d==2 (die beiden erfüllen **nicht** die b2 bei b==2, also müssen sie mit einer anderen b2 verknüpft sein).
Für die falsche b2 nehmen wir beispielsweise h==3. Dann fehlen uns für die falschen Lösungen mit b2 bei h==3 die Bedingung b6 und eine b5b. Also führt b2 immer noch zu keiner Lösung.
Nehmen wir eine b5a irgendwo dort, wo beispielsweise b3 nicht erfüllt wird, da wir immer noch nur eine b3 haben. Dann liege eine Hälfte der b5a bei c==1 und die andere irgendwo anders, z.B. bei e==4.
Jetzt suchen wir nach einer Alternative für b3. Sie soll entweder alle b1, alle b2, alle b4, alle b5 oder alle b6 ausschließen. Wir nehmen b5: b5a **und** b5b müssen ausgeschlossen, darum platzieren wir die eine b3 bei g==4 und die andere b3 bei e==2. Somit sind die beiden b3 ausgeschlossen, wir können sie also ignorieren.
Wir brauchen noch eine Alternative für b6. Die neue b6 soll beispielsweise b3 ausschließen, denn es gibt immer noch nur eine b3, die zur Lösung führen kann. Also liege b6 bei c==1.
Eine zweite falsche b4 können wir ebenfalls irgendwo dort platzieren, wo b3 ausgeschlossen wird. Z.B. bei a==4.
Somit haben wir:

(b==4||d==2||a==3)&&(b==2||h==3)&&(c==4||e==2||g==4)&&((d==1&&e==1)||(a==4))&&((c==1&&e==4)||(g==2&&h==4))&&(c==1||f==3)

⇔Implies[b≠4,d==2||a==3]&& Implies[b≠2,h==3]&& Implies[c≠4,e==2||g==4]&& Implies[d≠1||e≠1,a==4]&& Implies[c≠1||e≠4,g==2&&h==4]&& Implies[f≠3,c==1]

Nun ersetzen wir Zahlen und Variablen durch Personen und ihre Eigenschaften:
1 sei Bart Simpson, 2 sei Lisa Simpson, 3 sei Charles Montgomery Burns, 4 sei Homer Simpson;
a sei Farbtube, b sei Goldenmünze, c sei Zeitung, d sei Walkman;
e sei Kasino, f sei Fundbüro, g sei Geheimschrank, h sei Haus.

Unsere Logelei 8:

„Bart Simpson, Lisa Simpson, Charles Montgomery Burns und Homer Simpson hatten Glück heute und haben zufällig je eine Sache auf dem Weg nach Hause gefunden.
Wenn Homer keine Goldenmünze gefunden hat, dann hat Lisa den Walkman oder Charles eine Farbtube gefunden.
Wenn Lisa keine Goldenmünze gefunden hat, dann hat Charles seinen Fund nach Hause mitgebracht.
Wenn Homer keine Zeitung gefunden hat, dann hat Lisa ihren Fund im Kasino verspielt oder Homer den Fund in seinem Geheimschrank versteckt.
Wenn Bart keinen Walkman gefunden hat oder wenn er seinen Fund nicht im Kasino verspielt hat, dann hat Homer eine Farbtube gefunden.
Wenn Bart keine Zeitung gefunden hat oder wenn Homer seinen Fund nicht im Kasino verspielt hat, dann hat Lisa ihren Fund in den Geheimschrank gelegt und Homer seinen Fund nach Hause mitgebracht.
Wenn Charles seinen Fund nicht in das Fundbüro gebracht hat, dann hat Bart die Zeitung gefunden.
Nun wer hat was gefunden und was damit getan?“

5. Schlusswort

5.1. Zusammenfassung

Logeleien, mit denen ich mich beschäftigt habe, sind Textaufgaben, die logisches Denken erfordern. Sie werden zuerst vom Autor aufgebaut und mit Hilfe der logischen mathematischen Operationen in komplizierte Logeleien umgeschrieben.
Es ist möglich, Logeleien zu formalisieren. Um dies zu erreichen, können wir die gewöhnlichen Zahlen und Variablen, die wir aus der Mathematik kennen, gebrauchen, sowie auch Relationen und logischen Operationen „Wenn- Dann“, „Entweder- Oder“, „Oder“ und „Und“. Es gibt bestimmte Regeln, die uns helfen, die Operationen „Wenn- Dann“ und „Entweder- Oder“ zu vereinfachen und mit den Operationen „Und“ und „Oder“ aufzuschreiben. Die Zusammenhänge zwischen den beiden letzteren Operationen sind ebenfalls in der Arbeit dargestellt. Sie ermöglichen jedermann Logeleien zu lösen.

Die wichtigsten Regeln:

Wenn A, dann B	⇔	*A- Gegenteil oder B*
Entweder A oder B	⇔	*(A und B- Gegenteil) oder (A- Gegenteil und B)*
(A und B)- Gegenteil	⇔	*A- Gegenteil oder B- Gegenteil*
(A oder B)- Gegenteil	⇔	*A- Gegenteil und B- Gegenteil*
A und B und C	⇔	*C und B und A*
A oder B oder C	⇔	*C oder B oder A*
(A oder B)und (C oder D)	⇔	*(A und C)oder (A und D)oder (B und C)oder (B und D)*

Es gibt viele differente Logeleien, die auch auf verschiedene Arten gelöst werden können. Ich habe vier grundsätzlichen Lösungsverfahren entwickelt und erarbeitet, die in dieser Arbeit an mehreren Beispielen angewandt werden. Es ist möglich, mit Hilfe jedes der vier Verfahren jede Logelei zu lösen. Da aber der Aufwand für verschiedene Aufgaben verschieden ist, ist es sinnvoll, zuerst vor dem Lösen die Logeleien zu analysieren und sich für ein oder zwei bestimmte Lösungsverfahren zu entscheiden.

Das erste Lösungsverfahren basiert auf den Regeln aus der von mir erarbeiteten Theorie. Für das zweite Lösungsverfahren wird eine Tabelle mit allen möglichen Lösungen aufgestellt und es wird schrittweise nach der richtigen Lösung gesucht. Das dritte Verfahren ist etwas ähnlich wie das erste, es wird aber visualisiert dargestellt. Das vierte Lösungsverfahren eignet sich sehr gut für Logeleien mit den Relationen „Gleich“ und Operationen „Oder“, und ist sehr einfach zum Darstellen. Bei den einfacheren Logeleien ist es am schnellsten zum Anwenden.

Es ist ebenfalls möglich, Logeleien auf verschiedene Arten aufzustellen. Der einfachste Weg ist es, von der Lösung zur Logelei zu gehen. Wie es auch in den oben beigebrachten Regeln zu sehen ist, führen die Pfeile in beide Richtungen, was bedeutet, dass wir Logeleien nicht nur vereinfachen, sondern auch komplizierter machen können. So gehen wir von einer einfachen Form aus und gelangen zu einer schwierigen Aufgabe. Es gibt aber auch einige spezielle Regeln für den Aufbau von Logeleien, die eingehalten werden müssen. Ich habe grundsätzlich vier Arten vom Aufstellen von Logeleien betrachtet. Jede von ihnen hat ihre eigenen Spezialitäten, und es ist daher schwer zu sagen, welche von ihnen die beste ist.

Die formalisierten Logeleien kann man auch am PC im Programm *Mathematica* lösen. Im Unterschied zu Menschen gebraucht das Programm keine Regeln, um Logeleien zu lösen, sondern es spielt alle möglichen Lösungen durch und schaut welche von ihnen die Bedingungen erfüllen.

5.2. Fragen und Antworten

Am Anfang der Arbeit habe ich mir einige Fragen gestellt, die mich durch diese Arbeit geleitet haben. Am Schluss probiere ich, jede dieser Fragen zu beantworten:

- Ist Cemann mit Josefa verheiratet?

Ja, ist er. Die Antwort ist auf der Seite 35 zu finden.

- Heisst Demann Gustav?

Nein, heisst er nicht. Die Antwort befindet sich ebenfalls auf der Seite 35.

- Wie löst man solche Aufgaben?

Es gibt verschiedene Möglichkeiten, wie man solche Aufgaben lösen kann. Eine von ihnen ist, sie zu formalisieren (in Zahlen, Variablen, Relationen und Operationen umzuschreiben) und mit der Anwendung von Regeln zu vereinfachen.
Das Formalisieren von Logeleien und die Regeln dazu sind auf den Seiten 6 bis 19, die grundsätzlichen Lösungsverfahren auf den Seiten 20 bis 27 zu finden.

- Muss man gut logisch denken können, um Logeleien zu lösen?

Nein. Man muss nur die Logeleien formalisieren und die gegebenen Regeln anwenden können, um sie zu lösen. Es ist von einem grossen Vorteil, wenn man dabei etwas logisch denkt, denn es erspart zusätzlich Zeit, ist aber keine Voraussetzung.

- Ist es ein Zufall, dass diese Aufgabe nur eine Lösung hat, oder steckt etwas mehr dahinter? (Logelei 94)

Es könnte sein, dass es ein Zufall ist. Dann hat aber der Autor dieser Logelei Glück gehabt. Denn in dieser Logelei mit ihren Bedingungen gibt es 2^8=256 mögliche Lösungen.
Ich gehe davon aus, dass der Autor seine Logelei ähnlich wie ich aufgebaut hat. Und zwar so, dass sie genau eine Lösung hat.

- Kann ich auch eine Logelei aufstellen?

Ja, das kann ich auch. Der Beweis dazu befindet sich auf den Seiten 36 bis 49.

5.3. Zielsetzungen und Resultate

Zu meinen Zielen gehörten:

- Auseinandersetzung mit den Logeleien
- Selbständige Entwicklung und Erarbeitung der allgemeinen Theorie, die sich für die textgeschriebenen Logeleien eignen würde, ohne eventuell existierenden Theorien anzuschauen (fast 100% Eigenleistung)
- Entwicklung und Erarbeitung der Theorie, die auch meine MitschülerInnen nachvollziehen können
- Mögliche Lösungsmethoden von Logeleien zu erfinden und sie zu beschreiben
- Mögliche Aufstellungsmethoden von Logeleien zu erfinden, zu erarbeiten und meine eigenen Logeleien aufzubauen

Ich finde, dass ich meine Ziele erreicht habe.
Ich konnte auch unter anderem beweisen, dass es kein logisches Denken braucht, um Logeleien lösen zu können.

5.4. Quellenverzeichnis

- Das Buch „99 Logeleien von Zweistein“, Hoffmann und Campe, 6. Auflage, 1974
- Das lizenzierte PC- Programm *Mathematica*