# Grundlagen mehrwertiger formaler Logik, Vortrag bei MoMo

## 01 Worum`s mir geht:

Ich bemühe mich um den Aufbau mehrwertiger Logiken.
Das sind Logiken, in denen es nicht allein darum geht, ob Aussagen als "wahr" oder "falsch" zu bewerten sind. "Möglich" wäre ein dritter Wert und "unentscheidbar" ein vierter. Auch Anderes, wie Farben, könnte Werte zugeordnet bekommen.

Schon die Beschränkung auf diese Art Bewertung von Aussagen und die Beschränkung auf Aussagen überhaupt, erscheint mir zu eng. Ich meine, die aussagenlogischen Werkzeuge sind zu mehr zu gebrauchen, als nur zum Aussagen über Aussagen.
Und tatsächlich nutzen die Programmiersprachen, die ja alle auf die eine oder andere Art in der Aussagenlogik wurzeln, deren Formen und Regeln überwiegend nicht, um Aussagen zu beweisen, sondern um Computer dazu zu veranlassen z.B. farbige Zeichen auf einen Bildschirm zu malen, so wie dies auch in dem Textprogramm geschieht, mit dem dieser Vortrag erstellt wurde.

Wenn man sich also weniger um die Bedeutung von Sätzen kümmert, sondern mehr um die Formierungsregeln für die Anordnung von Zeichen, was ja neben der Bedeutung von Sätzen den anderen Teil der formalen Logik ausmacht, dann wird deutlicher, wie im Hintergrund die Maschinerie der Zeichenmechanik wirkt, sodaß Zeichen eben nicht nur im Stande sind, Bedeutung von Sätzen symbolisch zu binden, sondern auch eine CNC-Drehbank zu steuern oder neuronale Netze zu formulieren.

Meine Betonung liegt also nicht so sehr auf der Bedeutung von Werten und Sätzen ("Sätze" gleichbedeutend mit Behauptungen oderAussagen); nicht auf der Bedeutung einzelner Aussagen, sondern auf der Bereitstellung der formalen Mittel, mittels derer die Satzbedeutungen formal gefasst werden sollen.

Diesbezüglich redet man von Bedeutung als dem semantischen Aspekt und von den Regeln für die Zeichenanordnung als dem syntaktischen Aspekt, also von Semantik und Syntax. Diese Einordnungen gibt es auch für die Untersuchung des Aufbaus von natürlichen Sprachen und auch Formalsprachen (z.B. der mathematischen Formelsprache oder Programmiersprachen) ganz allgemein.

Außerdem kam meine Entscheidung, die formalen Mittel der Logik genauer zu untersuchen und die verschiedenen Inhalte von Aussagen zu vernachlässigen, auch aus einer Überlegung, wo es um die Form von Darstellung und ihre Wirkung auf das Dargestellte geht.

Man möchte ja meinen, es sei das Dargestellte, das bestimmt, wie die Darstellung auszusehen habe. Wenn aber die Mittel der Darstellung nicht im Stande sind, einen Gegenstand abzubilden, dann mag der Gegenstand noch so bedeutsam sein; es gibt keine Möglichkeit, ihn zu diskutieren.

Das heisst, daß die Gesetze der Darstellung nur die Darstellung von denjenigen Tatsachen erlauben, die sich den Gesetzen der Darstellung fügen. Sodaß es Tatsachen gibt, die zwar empfunden, aber nicht kommuniziert werden können, weil es für bestimmte Empfindungen eben keine Worte gibt, -höchstens die der Poesie. Wiewohl solche Empfindungen ebenso von Bedeutung für das Verstehen unserer Welt sind.

Man kann mit der formalen Logik eine Art Rechnen betreiben.
Damit das möglich wird, muß es bestimmte Regeln geben, die sich weniger nach den verschiedenen Inhalten von Aussagen richten, sondern mehr nach Formen, in die man die Aussagen bringen muß, um sie logisch verrechnen zu können: eben deren Syntax.

Ich beginne mit der Feststellung, daß es, genau genommen, "**die**" Logik garnicht gibt. Sondern es gibt zahlreiche Logik***en***, deren jede einen mehr oder weniger eigenen Begriffsapparat hat, zugeschnitten auf die jeweilige Aufgabe.

So gibt es eine Logik, die das Intervall von 0 bis 1 benutzt, die mehrwertige Logik des Herrn Kleene.
Da gilt "1" als "wahr" und "0" als "falsch". Dazwischen liegt in der Mitte "½" und das meint lt. Wikipedia: "weder wahr, noch falsch". Vielleicht ist damit sowas wie "vielleicht" gemeint.

Allen diesen Logiken ist gemeinsam, daß sie möglichst unzweideutig beschreiben sollen, was jeweilig für "sicher" oder, einfacher, für "wahr" gehalten wird. Damit sind die Gemeinsamkeiten aber auch schon so ziemlich erschöpft.

Die Logik, um die es mir geht, ist eine von den formalen und heisst "Junktorenlogik".
Da geht es um die Verknüpfung von vorausgesetzten Aussagen mittels verschiedener Verknüpfer, den "Junktoren" und Aussagen, die Schlüsse aus den verknüpften Voraussetzungen ziehen.

Die klassische Junktorenlogik kennt 16 Verknüpfer, Operatoren, die gemeinhin dafür zuständig sind, zwei Sätze auf bestimmte Weise miteinander in Beziehung zu bringen, sodaß es dadurch möglich wird, aus dem Zusammenhang eine Folgerung zu gewinnen.

Und sie kennt zwei Erscheinungsformen eines Satzes: Entweder affirmiert (der Satz oder Satzzusammenhang ist "wahr") oder negiert (der Satz oder Zusammenhang ist "falsch").
Die beiden Erscheinungsformen sprechen Bewertungen aus: Der Satz ist wahr und wird gebräuchlicher Weise kurz mit "1" bewertet oder er ist "falsch", Wert "0".

Beispiel:
Erster Satz "Es regnet auf die Strasse". Zweiter Satz: "Die Strasse wird nicht naß". Operator: "und".
Verknüpfung: "Es regnet auf die Strasse und die Strasse wird nicht naß."
Folgerung: Sofern die Strasse nicht überdacht war, ist der Satzzusammenhang falsch.

Meiner Ansicht nach kann und sollte man die klassische Junktorenlogik erweitern. Die Absicht des Vortrages ist es, einen ersten Eindruck zu geben, warum und wie.

Den Charakter aller Junktorenlogik, klassisch, wie mehrwertig, kann man in eine formelhafte Darstellung fassen, die durchaus an mathematische Formeln erinnert:

"**V** ⇒ **F**"

Zu lesen ist dies als: "Es gibt **V**oraussetzungen und die ziehen **F**olgen nach sich".
Darin steht "**V**" für Voraussetzung, "⇒" für "… ziehen nach sich …" und "**F**" für Folgen.
"**V**" ist das, was vorher war, "⇒" ist ein Operator, ein Verändern, das sodann, hervorgerufen von "**V**", geschieht und "**F**" ist dessen Resultat.

Man beachte, daß in dieser Formulierung kein direkt sinnlich erfahrbarer Sachverhalt vorkommt. "**V**","⇒" und "**F**" sind abstrakte "Hüllen", Schemata, in die man konkrete Dinge einfüllen kann.

Beispiel: "Aus einem Samen" (**V**oraussetzung) wächst (⇒) ein Baum (**F**olge)".

## 02 eine dreiwertige Wertetafel:

Die fundamentale grafische Figur, welche die Junktorenlogik nutzt und auf der sich alle logische Manipulation gründet, ist die der Wertetafel oder, wie sie in der klassischen zweiwertigen Logik genannt wird, die Wahrheitstafel.

Ich beginne aber gleich mit der Aufstellung des Beispiels für eine dreiwertige Tafel.

Zu diesem Zweck stellen wir uns einen Fluss vor, den es zu überqueren gilt.
Der Fluss umspült in seiner Mitte eine kleine Insel, die als Fundament für zwei Brücken dient. Die eine Brücke geht von unserem Ufer zur Insel hin und die zweite Brücke geht von der Insel weg zum anderen Ufer.

Aus der Ferne läßt sich schwerlich beurteilen, wie zuverlässig diese Brücken sind.
Jede der beiden Brücken könnte in gutem baulichem Zustand sein und damit sicher. Aber auch irgendwie brüchig, sodaß ihre unversehrte Überquerung fraglich erscheint. Oder ganz kaputt, sodaß man einbrechen würde und ertrinken müßte.
Wäre die erste Brücke auch sicher, wüßte man trotzdem noch nicht, ob man ans andere Ufer gelangt, denn der Zustand der zweiten Brücke wäre immer noch fraglich.

So sieht die zugehörige Wertetafel aus:

| ∧ ist und | Brücke e1 | ∧ | Brücke e0 | = | Flussüberquerung a |
|---|---|---|---|---|---|
| Fall 01 | intakt (2) | | intakt (2) | | gelingt (2) |
| Fall 02 | intakt (2) | | brüchig(1) | | vielleicht (1) |
| Fall 03 | intakt (2) | | kaputt (0) | | unmöglich (0) |
| Fall 04 | brüchig(1) | | intakt (2) | | vielleicht (1) |
| Fall 05 | brüchig(1) | | brüchig(1) | | vielleicht (1) |
| Fall 06 | brüchig(1) | | kaputt (0) | | unmöglich (0) |
| Fall 07 | kaputt (0) | | intakt (2) | | unmöglich (0) |
| Fall 08 | kaputt (0) | | brüchig(1) | | unmöglich (0) |
| Fall 09 | kaputt (0) | | kaputt (0) | | unmöglich (0) |

Man müße mit insgesamt folgenden Möglichkeiten rechnen:
Fall 01) Beide Brücken sind intakt. Dann kommt man gefahrlos ans andere Ufer.
Und, wie man sieht: "intakt" bekommt einen Wert zugeordnet, nämlich "2".
Ebenfalls bedeutet "2", daß die Überquerung gelingt.
"1" meint, die Brücke ist brüchig und "vielleicht kommt man rüber".
Mit "0" wird sowohl bewertet, daß die Brücke kaputt und auch, daß die Überquerung somit unmöglich ist.

Fall 02) Die erste Brücke ist sicher, aber die zweite scheint brüchig. Dann ist fraglich, ob man ans andere Ufer kommt.
Fall 03) Die erste Brücke ist sicher, aber die zweite ist kaputt. Dann wird man das andere Ufer sicher nicht erreichen.

Es könnte aber sein, daß schon die erste Brücke brüchig ist. Dann
Fall 04) wäre gleich klar, daß ihre sichere Überquerung fraglich erscheint und bestenfalls wäre die zweite Brücke sicher. So wäre, der ersten Brücke wegen, aber immer noch fraglich, ob man ans andere Ufer kommt.
Fall 05) Die erste Brücke ist brüchig und die zweite auch. Nun wäre die gefahrlose Überquerung erst recht unsicher, allerdings vielleicht immer noch möglich.
Fall 06) Die erste Brücke ist brüchig und die zweite ganz kaputt. Nix is mit Überquerung.

Schlimmstenfalls hat schon
Fall 07) die erste Brücke ein großes Loch. Dann nützt es auch nix, wenn die zweite Brücke ganz ist. Es ist von vornherein klar, daß man nicht auf die andere Seite kommt. Oder:
Fall 08) die erste Brücke ist kaputt und die zweite brüchig. Die zweite könnte man wohl notfalls überqueren, aber bis da kommt man ja garnicht.
Fall 09) Beide Brücken sind zerstört. Man sollte sich nach einer Fähre umsehen, selbst, wenn die ordentlich kostet.

Jeder der Fälle stellt eine These über die Flussüberquerung dar und zwar mit den Vorraussetzungen des Brückenzustands und der daraus folgenden Chance, ans andere Ufer zu kommen.
Weil diese Chance nicht von einer, sondern von zwei Brücken abhängig ist und für jede Brücke drei Zustände möglich sind, gibt es eben 9 Fälle, deren Konsequenzen man für`s Überqueren einzeln würdigen muß.

Jeder Einzelne dieser Fälle ist eine Junktion, bestehend aus Voraussetzungen (Zustand der Brücken), Operator (**∧**) und Folge (Überquerungsmöglichkeit).
Der Operator zeigt an, daß die, hier zwei, voraus gesetzten Sätze im Zusammenhang zu sehen sind und wie dieser Zusammenhang eine Schlußfolgerung erzeugt.
Die Sammlung aller dieser Fälle macht das Ganze der Wertetafel des Junktors aus.
Der Junktor sammelt also alle Möglichkeiten ein und jede einzelne Junktion ist eine der möglichen Realisierungen.

Zwischen natürlichsprachlichen Äußerungen und formallogischer Darstellung in einer Wertetabelle gibt es einen wichtigen Unterschied:
Wenn ich natürlichsprachlich äußere, was der Fall ist, ist damit durchaus nicht gesagt, was Alles dann nicht der Fall ist.
Ist die Brücke intakt, und Unversehrtheit ist ja eine mögliche Eigenschaft von Brücken, dann gibt es eine Unzahl von Eigenschaften der Brücke, über die gerade nicht gesprochen wird.
Die alle zu erwähnen, wäre unmöglich. Daher drückt natürliches Sprechen lieber Gegebenes aus, als das gerade nicht Gegebene.

Anders in der Wertetafel.
Hier wurden vorher alle Eigenschaften benannt, um die es geht. Die Liste der in Frage kommenden Eigenschaften ist daher normaler Weise kurz und jedenfalls endlich.
Wenn ich weiss, daß die Brücke intakt ist, dann kann ich auch aufzählen, was sie, im Rahmen der Wertetafel, dann nicht ist: Brüchig oder ganz kaputt.
Und wenn ich weiß, was die Brücke nicht ist, z.B. brüchig, dann weiß ich: Entweder ist sie intakt oder kaputt.

Jetzt benenne ich die einzelnen Teile, aus denen die Tafel aufgebaut ist. Der Teil um den es gerade geht, bleibt farbig, die anderen sind ausgegraut.
Aber Vorsicht: die Benennungen sind von mir. Sie dienen der inhaltlichen Bestimmung des Aufbaus dieser Wertetabelle und sind nicht allgemein übliche Bezeichnungen.

Den obersten, gelben Teil nenne ich den Kopf der Tafel.
Er enthält die Rubriken für die Sachverhalte, die dargestellt werden sollen: "**Brücke**" und "**Flussüberquerung**".
"**e1**" und "**e0**" sind Kurzbezeichnungen dafür, genauso, wie "**a**"

| ∧ ist und | Brücke e1 | ∧ | Brücke e0 | = | Flussüberquerung a |
|---|---|---|---|---|---|
| Fall 01 | intakt (2) | | intakt (2) | | gelingt (2) |
| Fall 02 | intakt (2) | | brüchig(1) | | vielleicht (1) |
| Fall 03 | intakt (2) | | kaputt (0) | | unmöglich (0) |
| Fall 04 | brüchig(1) | | intakt (2) | | vielleicht (1) |
| Fall 05 | brüchig(1) | | brüchig(1) | | vielleicht (1) |
| Fall 06 | brüchig(1) | | kaputt (0) | | unmöglich (0) |
| Fall 07 | kaputt (0) | | intakt (2) | | unmöglich (0) |
| Fall 08 | kaputt (0) | | brüchig(1) | | unmöglich (0) |
| Fall 09 | kaputt (0) | | kaputt (0) | | unmöglich (0) |

Kopf

**"und"**, Kurzzeichen: "**∧**" ist eine Vokabel der formalen Logik und kann, im Zusammenhang dieser Tabelle, durchaus wörtlich genommen werden. Wie in

| | Brücke e1 | ∧ | Brücke e0 | = | Flussüberquerung a |
|---|---|---|---|---|---|
| Fall 01 | intakt (2) | | intakt (2) | | gelingt (2) |
| Fall 02 | intakt (2) | | brüchig(1) | | vielleicht (1) |
| Fall 03 | intakt (2) | | kaputt (0) | | unmöglich (0) |
| Fall 04 | brüchig(1) | | intakt (2) | | vielleicht (1) |
| Fall 05 | brüchig(1) | | brüchig(1) | | vielleicht (1) |
| Fall 06 | brüchig(1) | | kaputt (0) | | unmöglich (0) |
| Fall 07 | kaputt (0) | | intakt (2) | | unmöglich (0) |
| Fall 08 | kaputt (0) | | brüchig(1) | | unmöglich (0) |
| Fall 09 | kaputt (0) | | kaputt (0) | | unmöglich (0) |

Fall 04: Brücke **e1** ist brüchig **und** Brücke **e0** ist intakt. **Also** ist die Flussüberquerung vielleicht möglich. "**=**" kann man sich als "**also**" vorstellen.

Den unteren Teil nenne ich "Körper".

| "und" | Brücke e1 | ∧ | Brücke e0 | = | Flussüberquerung a |
|---|---|---|---|---|---|
| Fall 01 | intakt (2) | | intakt (2) | | gelingt (2) |
| Fall 02 | intakt (2) | | brüchig(1) | | vielleicht (1) |
| Fall 03 | intakt (2) | | kaputt (0) | | unmöglich (0) |
| Fall 04 | brüchig(1) | | intakt (2) | | vielleicht (1) |
| Fall 05 | brüchig(1) | | brüchig(1) | | vielleicht (1) |
| Fall 06 | brüchig(1) | | kaputt (0) | | unmöglich (0) |
| Fall 07 | kaputt (0) | | intakt (2) | | unmöglich (0) |
| Fall 08 | kaputt (0) | | brüchig(1) | | unmöglich (0) |
| Fall 09 | kaputt (0) | | kaputt (0) | | unmöglich (0) |

Körper

Der ist wiederum aufgeteilt in einen "Eingangsteil"

| "und" | Brücke e1 | ∧ | Brücke e0 | = | Flussüberquerung a |
|---|---|---|---|---|---|
| Fall 01 | intakt (2) | | intakt (2) | | gelingt (2) |
| Fall 02 | intakt (2) | | brüchig(1) | | vielleicht (1) |
| Fall 03 | intakt (2) | | kaputt (0) | | unmöglich (0) |
| Fall 04 | brüchig(1) | | intakt (2) | | vielleicht (1) |
| Fall 05 | brüchig(1) | | brüchig(1) | | vielleicht (1) |
| Fall 06 | brüchig(1) | | kaputt (0) | | unmöglich (0) |
| Fall 07 | kaputt (0) | | intakt (2) | | unmöglich (0) |
| Fall 08 | kaputt (0) | | brüchig(1) | | unmöglich (0) |
| Fall 09 | kaputt (0) | | kaputt (0) | | unmöglich (0) |

Eingangsteil

und einen Ausgangsteil.

| "und" | Brücke e1 | ∧ | Brücke e0 | = | Flussüberquerung a |
|---|---|---|---|---|---|
| Fall 01 | intakt (2) | | intakt (2) | | gelingt (2) |
| Fall 02 | intakt (2) | | brüchig(1) | | vielleicht (1) |
| Fall 03 | intakt (2) | | kaputt (0) | | unmöglich (0) |
| Fall 04 | brüchig(1) | | intakt (2) | | vielleicht (1) |
| Fall 05 | brüchig(1) | | brüchig(1) | | vielleicht (1) |
| Fall 06 | brüchig(1) | | kaputt (0) | | unmöglich (0) |
| Fall 07 | kaputt (0) | | intakt (2) | | unmöglich (0) |
| Fall 08 | kaputt (0) | | brüchig(1) | | unmöglich (0) |
| Fall 09 | kaputt (0) | | kaputt (0) | | unmöglich (0) |

Ausgangsteil

Daher die Kurzbezeichnungen "**e1**" und "**e0**" (für eingang) und "**a**" (für ausgang).

Der Eingangsteil enthält die Zustände für die beiden Brücken samt in Klammern geschlossene Zahlenwerte und damit die Voraussetzungen "**V**".
Der Ausgangsteil enthält die Bewertungen zur Überquerbarkeit und damit eine Schlussfolgerung "**F**" nach der eingangs erwähnten Formel "**V** ⇒ **F**".
Diese Formel könnte man hier also spezifizieren zu **V**$_{(e1\wedge e0)}$ ⇒ **F**$_{(a)}$ .

In die ganze Wertetafel nun sind semantische Anteile (die Situation der Flußüberquerung) und ganz formale Anteile (Operatoren, Kürzel) eingegangen.
Läßt man die ganze semantische Bestimmung, die Brücken und die natürlich–sprachlichen Bewertungen weg, dann bleibt das dürre Gerippe eines auf Äußerste verkürzten Schemas als Junktor übrig.

| | Brücke e1 | ∧ | Brücke e0 | = | Flussüberquerung a |
|---|---|---|---|---|---|
| Fall 01 | intakt (2) | | intakt (2) | | gelingt (2) |
| Fall 02 | intakt (2) | | brüchig(1) | | vielleicht (1) |
| Fall 03 | intakt (2) | | kaputt (0) | | unmöglich (0) |
| Fall 04 | brüchig(1) | | intakt (2) | | vielleicht (1) |
| Fall 05 | brüchig(1) | | brüchig(1) | | vielleicht (1) |
| Fall 06 | brüchig(1) | | kaputt (0) | | unmöglich (0) |
| Fall 07 | kaputt (0) | | intakt (2) | | unmöglich (0) |
| Fall 08 | kaputt (0) | | brüchig(1) | | unmöglich (0) |
| Fall 09 | kaputt (0) | | kaputt (0) | | unmöglich (0) |

**Wertetafel mit semantischer Bestimmung**

| e1 | ∧e0 | = a |
|---|---|---|
| 2 | 2 | 2 |
| 2 | 1 | 1 |
| 2 | 0 | 0 |
| 1 | 2 | 1 |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 2 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

**"und"-Schema**

Das logische "**∧**", was im "und"-Schema gleich auffällt, nimmt immer den kleineren Eingangswert und ordnet ihn als Ausgangswert zu. Oder, wenn beide Eingangswerte gleich sind, diesen.

Was soll das? könnte man fragen. Ohne den konkreten Bezug macht die ganze Tafel keinen Sinn! Doch.

## 03 Bedeutung und Form:

Die Tafel mit der semantischen Bestimmung besteht aus "externen" Anteilen, den Bedeutungen der Sätze, eben der semantischen Bestimmung. Und "internen" Anteilen, den Zeichenbeziehungen in der Wertetafel oder im logischen Schema untereinander.
Der externe Anteil kommt aus der Welterfahrung des Tabellenaufstellers. Seine Erfahrung versichert ihn z.B. der Tatsache, daß man ertränke, würde man eine komplett kaputte Brücke überqueren wollen.
Die internen Anteile betreffen das Formale, das die Welterfahrung logisch rahmen soll, nämlich, wie eingangs erwähnt, die Syntax.

Die Erfahrung zwingt den Aufsteller, hat er den natürlichsprachlichen Wertungen erst mal Zahlenwerte zugeordnet, die Zahlenwerte, seiner Erfahrung gemäß, auch zu benutzen, damit er keine Inkonsistenzen in der formalen Darstellung seiner Ansichten erzeugt:

"0 0 | 2" ist falsch, wenn "0 0" für "Brücken kaputt" stehen soll und "2" für "Übergang ans andere Ufer gelingt".

Dabei mögen Ansichten über Sachverhalte durchaus strittig oder unsicher sein, nicht so evident wie im Beispiel hier. Aber jedenfalls wird man durch die Kodifizierung gezwungen, sich im Hinblick auf die Voraussetzungen, die man gemacht hat, als auch auf die Schlüsse, die man ziehen will, klar zu äußern.
Und dies ist die Klärung, welche die Bedeutung in eine Wertetabelle mit semantischer Bestimmung einbringt.

Das gilt nicht mehr für eine Wertetabelle ganz ohne externe Anteile.
Hier wird der Blick auf ganz andere Fragen, nämlich kombinatiorischer Art, gelenkt und die betreffen das interne Zusammenspiel der mittels Zahlen bezeichneten Werte.

Die Wertetafel, bzw. das Schema, ist hier zweistellig, was sich nach der Anzahl der hier gemachten Eingangsvoraussetzungen richtet: Es gibt zwei Brücken. Wären es drei Brücken, wäre die Tafel dreistellig.
Außerdem ist die Wertetafel dreiwertig. Es gibt die Werte 2, 1 und 0. Die werden zwar ein- wie ausgangsseitig verschieden benannt, aber z.B. das ausgangsseitige "vielleicht" schwingt schon eingangsseitig im "brüchig" mit.
Hinter den verschiedenerlei Bennenungen steht ein durchaus Ordnendes, semantisches Ganzes.

Außerdem dürfen Wertezusammenstellungen im Eingangsbereich sich weder mehrfach wiederholen, noch weggelassen werden:

"0 1 | 0" darf nur einmal vorkommen. Schon garnicht mehrfach und mit verschiedenen Schlussfolgerungen, sodaß als Ergebnis beim nächsten Mal ein anderer Wert zugeordnet würde.
Die Tafel muß neun verschiedene Fälle unterscheiden. Acht wären einer zu wenig und bei zehn Fällen wäre einer doppelt aufgetreten.

Außerdem möchte man wissen, wieviele Junktoren es eigentlich geben kann. Der im Beispiel gegebene "**∧**"–Junktor ist einer davon.

Dies hängt an der Anzahl der Ausgänge, hier 9 und den 3 prinzipiell möglichen Ausgangswerten pro Fall.
Demzufolge gibt es hier 3 hoch 9 mögliche Junktoren, stolze 19 683 Stück.
Wäre die Tafel zweistellig und zweiwertig gewesen, hätte man nur 16 verschiedene Junktoren benennen müssen.
Schon die Erweiterung der Wertigkeit der Logik um lediglich einen weiteren Wert, läßt die Anzahl der Junktoren mehr als sprunghaft ansteigen.

Da die klassische Logik nur zweiwertig ist, spielen zweistellige, zweiwertige Wertetabellen, dort "Wahrheitstafeln" genannt, eine besondere Rolle und jeder der 16 Junktoren hat auch tatsächlich einen eigenen Namen.
Allerdings sind nur wenige dieser Junktoren wichtig. Im engeren Sinne nur zwei: der Junktor des "logischen und" und der des "logischen oder".

Wertetabellen ohne externe Bestimmung sind dann logische Schlussschemata. Sie geben "Hüllen" für logisch überprüfbares Argumentieren ab.

Interessant wird ein Schema, wenn es auf Wirklichkeit trifft, die dieser Logik gehorcht, wie im Fall des Beispiels der zwei Brücken.
Die Schematisierung bringt dann eine drastische Reduktion an Informationen, die man im Gedächtnis behalten muß und die Frage, wie Wirklichkeit wohl beschaffen sein muß, damit ein bestimmter Junktor einen ihrer Sachverhalte zu beschreiben kann.

Man greife einen beliebigen Fall heraus, z.B. "0 2 | 0".
Die Regel, die den zugehörigen "**∧**"–Junktor beherrscht ist die: Nimm den kleinsten Eingangswert, hier "0". Der ist gleich dem Ausgangswert: "0". Falls beide Eingangswerte gleich lauten, nimm das.

Um die Semantik bereichert, ergibt sich so die Gesamtaussage

Fall 07: **Brücke e1** ist kaputt (0) **und Brücke e0** ist intakt (2) | **Die Überquerung a ist** unmöglich (0)

Oder

Fall 02: **Brücke e1** ist intakt (2) **und Brücke e0** ist brüchig(1) | **Die Überquerung a ist** vielleicht(1) **möglich**

Wie schon erwähnt, es gibt noch eine Menge anderer Junktoren.
Greifen wir den anderen sehr Gebräuchlichen heraus, den Junktor "oder auch", Kurzzeichen "**∨**". Wie läßt sich der auf das Brückenbeispiel anwenden?

Dazu verändern wir die Situation für die Brücken.
Die sind nun nicht hintereinander angeordnet, sondern gehen parallel über den Fluß und es gibt keine Insel. Das bedeutet: Mehr Verläßlichkeit für die erfolgreiche Flußüberquerung.
War die im Falle des "und", Zeichen: "**∧**" schon nicht mehr möglich, wenn auch nur eine der Brücken komplett zerstört war, sagen wir jetzt: "Macht nix, nehmen wir die andere!"
Nur wenn die auch ganz kaputt ist, kommen wir nicht auf die andere Seite.

Das bringen wir in unsere Wertetafel ein und verändern die dann so:

| ∨ (oder) | Brücke e1 | ∨ | Brücke e0 | = | Flussüberquerung a |
|---|---|---|---|---|---|
| Fall 01 | intakt (2) | | intakt (2) | | gelingt (2) |
| Fall 02 | intakt (2) | | brüchig(1) | | gelingt (2) |
| Fall 03 | intakt (2) | | kaputt (0) | | gelingt (2) |
| Fall 04 | brüchig(1) | | intakt (2) | | gelingt (2) |
| Fall 05 | brüchig(1) | | brüchig(1) | | vielleicht (1) |
| Fall 06 | brüchig(1) | | kaputt (0) | | vielleicht (1) |
| Fall 07 | kaputt (0) | | intakt (2) | | gelingt (2) |
| Fall 08 | kaputt (0) | | brüchig(1) | | vielleicht (1) |
| Fall 09 | kaputt (0) | | kaputt (0) | | unmöglich (0) |

**Wertetafel mit semantischer Bestimmung**

| e1 | ∨e0 | = a . |
|---|---|---|
| 2 | 2 | 2 |
| 2 | 1 | 2 |
| 2 | 0 | 2 |
| 1 | 2 | 2 |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 2 | 2 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

**"oder auch"-Schema**

| ∧ ist und | Brücke e1 | ∧ | Brücke e0 | = | Flussüberquerung a |
|---|---|---|---|---|---|
| Fall 01 | intakt (2) | | intakt (2) | | gelingt (2) |
| Fall 02 | intakt (2) | | brüchig(1) | | vielleicht (1) |
| Fall 03 | intakt (2) | | kaputt (0) | | unmöglich (0) |
| Fall 04 | brüchig(1) | | intakt (2) | | vielleicht (1) |
| Fall 05 | brüchig(1) | | brüchig(1) | | vielleicht (1) |
| Fall 06 | brüchig(1) | | kaputt (0) | | unmöglich (0) |
| Fall 07 | kaputt (0) | | intakt (2) | | unmöglich (0) |
| Fall 08 | kaputt (0) | | brüchig(1) | | unmöglich (0) |
| Fall 09 | kaputt (0) | | kaputt (0) | | unmöglich (0) |

**Wertetafel mit semantischer Bestimmung**

| e1 | ∧e0 | = a |
|---|---|---|
| 2 | 2 | 2 |
| 2 | 1 | 1 |
| 2 | 0 | 0 |
| 1 | 2 | 1 |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 2 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

**"und"-Schema**

Auf Brücken an`s andere Ufer zu kommen, die in der junktorellen "∨"-Beziehung stehen, sind die Chancen ungleich besser, was sich im direkten Vergleich beider Wertetabellen sofort erschließt:

| "Oder auch (∨)" | "und(∧)" |
|---|---|
| In 5 Fällen, 01 bis 04 und 07 gelingt die Überquerung sicher. | Nur im einen Fall 01 ist sie sicher möglich. |
| In 3 Fällen, 05, 06 und 08 ist sie immerhin vielleicht möglich. | In 3 Fällen, 02, 04 und 05 ist sie immerhin möglich. |
| Nur im einen Fall 09 ist sie aussichtslos. | Aber 5 Fällen, 03, 06 und 07 bis 09, geht`s garnicht. |

Für den Operator **"und(∧)"** ließ sich eine Kurzregel aufstellen: Nimm den **kleineren** Eingangswert und mach den zum Ergebnis.
Und als Kurzregel für den Operator **"Oder auch (∨)"** gilt : Nimm den **größeren** Eingangswert und mach den zum Ergebnis.
Und für beide Operatoren gilt: Sind beide Eingangswerte gleich groß, nimm als Ergebnis dies.

Im Vergleich der Schemata:

| e1 | ∨e0 | = a |
|---|---|---|
| 2 | 2 | 2 |
| 2 | 1 | 2 |
| 2 | 0 | 2 |
| 1 | 2 | 2 |
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 2 | 2 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

**"oder auch"-Schema**

| e1 | ∧e0 | = a |
|---|---|---|
| 2 | 2 | 2 |
| 2 | 1 | 1 |
| 2 | 0 | 0 |
| 1 | 2 | 1 |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 2 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

**"und"-Schema**

## 04 Die Negation der Negation:

Eine wichtige Regel der klassischen, zweiwertigen Logik gilt der Negation der Negation. Die werde ich jetzt näher erläutern und dann zeigen, wie man sie für mehrwertige Logiken erweitern kann.

Als Beispiel nehmen wir diesmal nur eine Brücke allein und bedienen uns der klassischen zweiwertigen Logik. Ein dritter Wert fällt somit flach. Damit bleiben "Es ist so, daß die Brücke intakt ist" und "Es ist nicht so."

Sodann negieren wir:
1. Negation: "Die Brücke ist nicht intakt."
2. Negation: "Es ist nicht so, daß die Brücke nicht intakt ist."

Was heisst: "Die Brücke ist sehr wohl intakt." Und was damit dem Ergebnis einer einfachen Affirmation gleicht.

Dabei ist zu beachten: Die erste Negation negiert den Inhalt einer Aussage. Sie negiert den Inhalt der Aussage "Die Brücke ist intakt.".
Die zweite Negation negiert keinen Inhalt, sondern den Wert einer Aussage über die Brücke: "Es ist nicht so, daß die Brücke nicht intakt ist."

Eine zweite Negation negiert eine Aussage nicht allein, sondern bezüglich des Inhalts der vorherigen Negation. Eine dritte Negation würde die zweite Negation negieren, usw.
Dies hieße hier: "Es ist nicht so (dritte Negation), daß es nicht so ist (zweite Negation), daß die Brücke nicht intakt ist (erste und inhaltliche Negation)." Kurz gesagt: "Die Brücke ist nicht intakt."
Die Anreihungen der Negationen lassen im Ergebnis den Inhalt zwischen Negation und Affirmation oszillieren.

Wir wollen aber wieder alle Möglichkeiten darstellen, um eine vollständige Tabelle zu bekommen. Und dann reicht es nicht, sich nur um die Negation zu kümmern, sondern, wir müssen auch die Affirmation hinzu nehmen.

Daher ergibt sich folgendes Bild der tabellarischen Gesamtheit:

| | | | | | | |
|---|---|---|---|---|---|---|
| 1) die Affirmierung | des Affirmierten : | "Es ist so, | daß es so | ist, daß die Brücke intakt ist". | Gleich zu setzen mit | "Es ist so, daß die Brücke intakt ist." |
| 2) die Affirmierung | des Negierten : | "Es ist so, | daß es so nicht | ist, daß die Brücke intakt ist". | Gleich zu setzen mit | "Es ist so nicht, daß die Brücke intakt ist". |
| 3) die Negierung | des Affirmierten : | "Es ist so nicht, | daß es so | ist, daß die Brücke intakt ist". | Gleich zu setzen mit | "Es ist so nicht, daß die Brücke intakt ist". |
| 4) die Negierung | des Negierten : | "Es ist so nicht, | daß es so nicht | ist, daß die Brücke intakt ist". | Gleich zu setzen mit | "Es ist so, daß die Brücke intakt ist." |

Die Mechanik, die da wirkt, läßt sich am Besten mit Pfeildiagrammen darstellen.

Zunächst das einfache Affirmieren …

```
des Affirmierten (1)──►(1): das Affirmierte bleibt affirmiert. Es hatte den Wert 1 und behält ihn.
des Negierten    (0)──►(0): das Negierte    bleibt negiert.    Es hatte den Wert 0 und behält ihn.
```

Nun das einfache Negieren …

```
des Affirmierten (1)╲╱(0): das Affirmierte wird Negiertes.   Was den Wert 1 hatte, bekommt den Wert 0.
des Negierten    (0)╱╲(1): das Negierte    wird Affirmiertes.Was den Wert 0 hatte, bekommt den Wert 1.
```

Wie man sieht, kreuzen sich hier die Pfeile. Damit zeigen sie die Mechanik des Negierens.
Sie zeigen im gleichen Diagramm, was man tun muß, wenn man "1" negieren will und was man tun muß, wenn man "0" negieren will: man muß die beiden Werte vertauschen.

Jetzt wenden wir die Pfeilmechanik jeweils zweimal hintereinander an. Zunächst das zweimalige Affirmieren:

```
               erste Affirmierung                      zweite Affirmierung
Das Affirmierte   (1)──►(1)  bleibt affirmiert im ersten Schritt (1) ──►(1) und auch im zweiten.
Das Negierte      (0)──►(0)  bleibt negiert    im ersten Schritt (0) ──►(0) und auch im zweiten.
```

Klar: Alles bleibt, wie es war.

Aber das mehrmalige Negieren:

```
                  erste  Negierung                 zweite  Negierung
das Affirmierte  (1)╲╱(0)  wird zum Negat      (0)╲╱(1) und wieder zum Affirmat.
das Negierte     (0)╱╲(1)  wird zum Affirmat   (1)╱╲(0) und wieder zum Negat.
```

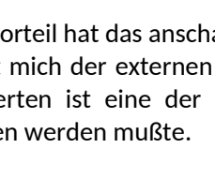
… bewirkt ein Oszillieren zwischen Negieren und Affirmieren.

Welchen Vorteil hat das anschauliche Umwerten mittels Pfeildynamik?
Es enthebt mich der externen Deutung des Umwertens. Ich muß nicht nach einer Bedeutung fahnden, die zum Umwerten paßt. Das Umwerten ist eine der internen Regeln zum Umgang mit Zeichen geworden, ohne daß den Zeichen eine Bedeutung angemessen werden mußte.

Und das macht die Möglichkeit des Umwertens in mehrwertigen Logiken auf.
Wobei das zweiwertige Umwerten als Sonderfall in dreiwertige Logiken vollständig übernommen werden kann, das dreiwertige Umwerten in vierwertige Logiken, etc…
Das Umwerten wird das, was man zum Zusammenfassen mehrwertiger Logiken in einem Gesamtsystem mehrwertiger Logik insgesamt benötigt: Das Umwerten wird auf– und abwärtskompatibel zwischen höher- und niederwertiger Logik bis hinunter zur klassisch-zweiwertigen.

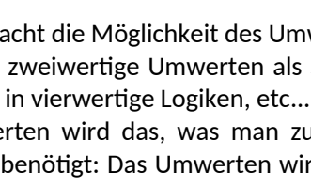
Es gibt nun mehrere Möglichkeiten, Affirmation wie Negation und Negation der Negation in dreiwertige Logik und noch höher wertige zu übernehmen. Ich werde eine ganz einfache zeigen.

Affirmation in einer dreiwertigen Logik sieht so aus:

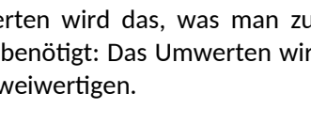


Wegen der Erweiterung auf drei Werte gibt es ausser den Werten "1" und "0" nun noch den Wert "2". Das Affirmieren selbst beläßt alle dies Werte, genau wie dies auch in der zweiwertigen Logik war.
Was der Wert "2" bedeuten könnte, ist für den Blickwinkel des internen Umgangs mit den Zeichen unerheblich. Da reicht die Pfeilmechanik.
Falls man auf einen Sachverhalt stößt, der den dritten Wert benötigen könnte, damit seine Bedeutung formalisiert werden kann, dann jedenfalls steht der dritte Wert zu Verfügung.

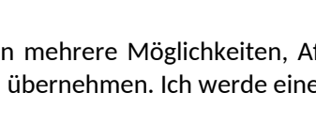
Ansonsten ist die klassische Negierung mit dem Ergebnis "Affirmierung durch doppeltes Negieren" dreifach enthalten, nämlich so,



oder so,



aber auch so:



Jeweils einer der Werte bleibt dabei immer affirmiert: der oben, der in der Mitte oder der unten.

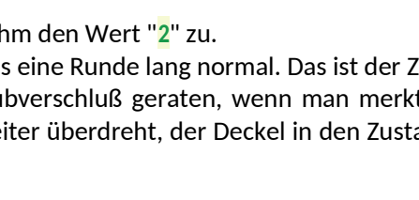
Wegen der Dreiwertigkeit dieser Logik kommen nun aber noch zwei neue Ausdrucksformen von Negierung dazu. Um die plausibel zu machen, knüpfe ich an eine Situation an, die sich nachvollziehen läßt (hoffentlich):

Wer das Gewinde des Schraubverschlusses einer Flasche überdreht, kann folgende Beobachtung machen:
Nachdem man die Flasche zugedreht hat und dann aber noch weiterdreht, wehrt sich der Verschlussdeckel vom Gewinde gerissen zu werden, indem er den Verschluss wieder ein Stück hoch springen läßt.

Dies soll der Anfangszustand sein und ich ordne ihm den Wert "2" zu.
Dreht man dann wieder in Richtung "zu", geht das eine Runde lang normal. Das ist der Zwischenzustand mit dem Wert "1".
In den dritten Zustand, Wert "0", ist der Schraubverschluß geraten, wenn man merkt, daß er dem Überdrehen einen gewissen Widerstand entgegensetzt, sodaß, wenn man weiter überdreht, der Deckel in den Zustand "2" zurück springt und die Dynamik von vorne los geht.

Nun das Ganze als Pfeilmechanik, die ja sämtliche mögliche Anfänge samt Folgen in einer Darstellung zusammen faßt:

```
Anfangszustand:  Überdreht
Zwischenzustand: eine Drehung weiter
Dritter Zustand: Drehung gehemmt
```



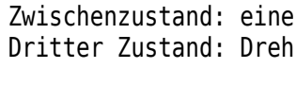
Weil ja das Pfeildiagramm alle Zustände umfassen soll, müssen mittelgraue und hellgraue Zeile, also auch die anderen möglichen beiden Anfangszustände mit ihren Weiterungen enthalten sein.

Wir verfolgen das Geschehen in der dunkelgrauen Zeile: Die beginnt im überdrehten Zustand.
Durch einmaliges Weiterdrehen springt der Schraubverschluss ganz nach oben.
Abermaliges Drehen erzeugt den gehemmten Zustand kurz vorm Überdrehen und
durch Weiterdrehen kommt der Verschluss wieder zurück in den Anfangszustand der Überdrehung.

Diese im Ganzen des Diagramms simultan dargestellten Zustände lassen uns sehen, wie die Pfeilmechanik das jeweilige Format an die richtige Stelle in der jeweiligen Zeile rückt. Die Pfeilmechanik stellt dar, daß es in einer dreiwertigen Logik auch eine dreifache Negation gibt, die dem Ergebnis der einfache Affirmation gleicht.

Außerdem:
Spiegelt man diese Negierung, indem man den langen Pfeil, statt von oben nach unten, von unten nach oben zeigen läßt und schließt sie an die erste Negierung an, verhält sich das Negieren insofern fast klassisch, daß sich nach dem zweiten, gespiegelten Negieren der affirmierte Zustand einstellt.

So sieht das aus:



Es entspräche einem Vor– und dann Zurückdrehen des Schraubverschlusses.

Insgesamt gesehen gibt es also in der zweiwertigen Logik ein Affirmieren und ein Negieren, in der dreiwertigen Logik drei Negierungen, die dem zweiwertigen Negieren in etwa entsprechen und zwei neue Sorten Negierung, wovon die eine die Spiegelung der anderen ist. Außerdem gibt es natürlich, wie auch in der zweiwertigen Logik, wieder die vollständige Affirmierung.

Damit hat die zweiwertige Logik zwei Pfeilmechaniken.
Die dreiwertige Logik mit Affirmierung und Negierung hat dagegen sechs Pfeilmechaniken:
Drei Sorten klassisches Negieren, zwei Sorten Neues, nämlich Affirmation durch Negation der Negation der Negation und wieder die gesamte Affirmierung aller Werte.

In einer vierwertigen Logik hätte man 23 Sorten Negierung und abermals die eine Affirmierung.
Für jede mehrwertige Logik läßt sich die Anzahl der Negierungen, plus immer der einen Affirmierung, leicht ausrechnen. Man geht da nach dem Prinzip der Fakultätsrechnung vor:

Für die klassische zweiwertige Logik, Logik$_2$, berechnet man die eine Art Negieren plus Affirmieren so: 1x2=2. Man erhält als Anzahl der Bewertungen: 2 Bewertungen (Affirmierung plus Negierung).

Nun die Anzahl der Pfeilmechaniken für die Bewertungen in einer dreiwertigen Logik, Logik$_3$: 1x2x3=6.
Dies sind: die Affirmierung, drei Negierungen, die sich so verhalten, wie in Logik$_2$, wo das doppelte Negieren auf`s einfache Affirmieren führt und die neuen Negierungen, wo dreimal ausgeführt werden muß, um die Affirmierung zurück zu erhalten.
Also insgesamt 6 Pfeilmechaniken.
In Logik$_4$ gibt es schon 1x2x3x4=24 Pfeilmechaniken. In Logik$_5$ 1x2x3x4x5=120 Pfeilmechaniken, in Logik$_6$ 1x2x3x4x5x6=720 Pfeilmechaniken und so weiter.

Wie zu sehen, kann man leicht ein allgemeines Bildungsgesetz ablesen:
Für eine Logik$_n$ mit n Bewertungen läßt sich die Anzahl der Pfeilmechaniken (also immer die eine Affirmierung plus alle Negierungen der Logik$_n$) konstruieren, indem man das Produkt von 1 bis n bildet und dann ausrechnet, wie: 1x2x3x4x5x6 … xn.
Um das Produkt nicht explizit hinschreiben zu müssen, weil man ja weiss, daß der nächst folgende Faktor immer um 1 größer ist, schreibt man kurz: n! (n mit Ausrufezeichen dahinter) und sagt dazu "n Fakultät".
Also 2 Fakultät: 2! =1x2=2, 3 Fakultät: 3! =1x2x3=6, 4 Fakultät: 4! =1x2x3x4=24, usw… bis n Fakultät: n! = 1x2x3x4x5 … xn= … wie groß diese Zahl auch immer sein mag.

## 04a Aber nun noch was für die, die`s genauer wissen wollen: Eine Umrechnung von "und" nach "oder"

Ich habe in **"01 Worum`s mir geht:"** behauptet, man könne mit der formalen Logik eine Art Rechnen betreiben. Was man da so alles machen kann, das möchte ich in diesem Abschnitt beispielhaft darstellen.

Man kann nämlich beispielsweise das logische "und" in ein logisches "oder auch" umrechnen.
Dazu beginne ich mit den beiden Junktoren "und(**∧**)" und "oder auch(**∨**)" der klassischen Logik$_2$ und werde dann die Aufwärtskompatibilität der Formen in die mehrwertige Logik$_3$ demonstrieren.
Außerdem werde ich ab jetzt eine semantische Ausdeutung völlig vernachlässigen, um zu zeigen, daß man allein mit syntaktischem Vorgehen auskommt. Es werden daher nur noch die junktorenlogischen Schemata benutzt.

Deren zweiwertige Wahrheitstafeln sehen so aus:

| e1 | ∧e0 | = a. |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

**"und"**

| e1 | ∨e0 | = a |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 0 |

**"oder auch"**

Anscheinend sind die beiden Tafeln irgendwie symetrisch: Dreht man die "**a**"-Spalte des "**∧**", "10 00" von oben nach unten, erhält man "00 01". Und tauscht man danach noch die "0"- Werte um in "1"- Werte und den "1"- Wert um in den "0"- Wert, dann erhält man die "**a**"-Spalte des "**∨**": "11 10".

So leicht darf man sich`s aber nun nicht machen. Auch deshalb nicht, weil ja das Ganze später in Logik$_3$ Wertetafeln transponiert werden soll. Man muß sich also an die Umformungsvorschriften der Junktorenlogik halten.

Beim Aufstellen der Wertetafel wird man eine bestimmte Zeilenfolge einhalten. Meine richtet sich von oben nach unten und von den höchsten zu den niedrigsten Eingangswerten. Einmal festgelegt, wird man auch die anderen Wertetafeln, die man vergleichen will, so ordnen. Ansonsten gibt es aber kein Vorgehen zur räumlichen Umordnung von Wertetafeln.
Es gibt nur das Vertauschen von Eingangswerten und das grundsätzlich wählbare Zuordnen von Ausgangswerten.

Unter diesen Vorschriften sieht die Transformation vom "**∧**" zum "**∨**" so aus:

Start:

| e1 | ∧e0 | = a |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

1.Schritt:

| $\overline{e1}$ | ∧$\overline{e0}$ | = a |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Vertauschen der Werte im Körper: Aus "1" wird "0" und aus "0" wird "1".
Der Eingang wurde somit komplett negiert.
Beibehalten wird wird die Zuordnungsvorschrift für "**∧**": Als Ergebnis in "**a**" wird der kleinere Eingangswert genommen, falls beide Werte verschieden sind.
Nebenbei: Damit gleicht die Spalte unter "**a**" der des "weder/noch" Operators.
Im Kopf wird das Negieren von **e1** und **e0**, wie man sieht, durch einen Oberstrich gekennzeichnet. Ich habe ihn rot hervor gehoben.

2.Schritt:

| $\overline{\overline{e1}\wedge\overline{e0}}$ | | = a |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Am Ausgang "**a**" werden die Werte vertauscht. D.h., daß nun der Ausgang negiert wurde.
Dies wird mit einem weiteren Strich über dem ganzen Eingang $\overline{e1\wedge e0}$ gekennzeichnet, um auszudrücken, daß diesmal der Eingangsteil des Kopfes als Ganzes negiert wurde, woraus die Umwertung von "**a**" kam.

Nunmehr entspricht die Spalte unter "**a**" der Ergebnis- (oder Zuordnungsspalte) von "**∨**".

Wie läßt sich dieses Verfahren jetzt auf Logik$_3$ anwenden?
Klar ist schon mal, daß das Negieren irgendwie anders gekennzeichnet werden muß. Schließlich gibt es in Logik$_3$ fünf Arten des Negierens, aber Überstreichen ist nur eine Möglichkeit um das Negieren zu kennzeichnen.
Also musste ich mir selber was ausdenken.

Meine Kopfformel der Wertetafel sieht nun so aus:

$$\left(\overset{012}{e1}\wedge\overset{012}{e0}\right)^{012}=a$$

Und die Interpretation:
Man denke sich über der Zeile "012" eine weitere Zeile, wo von links nach rechts der Wertevorrat heruntergezählt wird; etwa so:

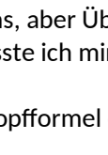
210
012

Unter der grauen 2 im grauen Feld steht die rote 0.
Das heisst, überall im Körper der Wertetafel, wo der Wert 2 auftaucht, muß er durch 0 ersetzt werden.

Unter der grauen 1 im grauen Feld steht die rote 1.
Die 1 bleibt.

Unter der grauen 0 im grauen Feld steht die rote 2.
Überall im Körper, wo der Wert 0 auftaucht, soll er durch 2 ersetzt werden.

Dies erinnert nicht von ungefähr an die Pfeilmechanik:



Und nun wenden wir dies auf die Schemata zur Logik$_3$ an:

Start:

| 210 | 210 | 210 |
|---|---|---|
| e1 | ∧e0 | = a |
| 2 | 2 | 2 |
| 2 | 1 | 1 |
| 2 | 0 | 0 |
| 1 | 2 | 1 |
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 2 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

**"und"-Schema**

Hier habe ich auch für die Start-Wertetafel des "**∧**" die Formate von "**e1**", "**e0**" und "**a**" über die Eingänge und den Ausgang drüber geschrieben. Denn, explizit angegeben, handelt es sich dabei ja um jeweils reine Affirmate, also keinerlei Zeichenaustausch im Körper der Wertetafel.

1.Schritt:

| 012 | 012 | 210 |
|---|---|---|
| e1 | ∧e0 | = a |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 2 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| 1 | 2 | 1 |
| 2 | 0 | 0 |
| 2 | 1 | 1 |
| 2 | 2 | 2 |

**Zeichentausch e1∧e0**

Der 1. Schritt ändert die Sachlage insofern, daß am Eingang die Zeichen gemäß der Formatierungsvorschrift, der Pfeilmechanik, getauscht werden; eben 2 zu 0 und 0 zu 2, während der 1 nichts geschieht.
Für den ganzen "und" Zusammenhang bleibt das Format aber das Affirmat und damit gilt nach wie vor: Der kleinere Wert erscheint am Ausgang, wenn nicht beide Werte gleich sind.
Und ein "weder/noch" zeigt sich nicht als am Ausgang als Zwischenschritt. Was daran liegt, daß, wie schon in "**03 Bedeutung und Form"** angesprochen, eine Riesenmenge mehr an neuen Junktoren entstanden war.

2.Schritt:

| (012 | 012) 012 | | 210 | 012 |
|---|---|---|---|---|
| (e1 | ∧e0) | = | a(∧) | zu a(∨) |
| 0 | 0 | | 0 | 2 |
| 0 | 1 | | 0 | 2 |
| 0 | 2 | | 0 | 2 |
| 1 | 0 | | 0 | 2 |
| 1 | 1 | | 1 | 1 |
| 1 | 2 | | 1 | 1 |
| 2 | 0 | | 0 | 2 |
| 2 | 1 | | 1 | 1 |
| 2 | 2 | | 2 | 0 |

**Zeichentausch a**

Nun noch der Zeichentausch für die Gesamtheit der Formel und "**a**" hat im Ergebnis die "oder"-Zuordnung.

## 05 Schlussbetrachtung:

Meiner Ansicht nach bereichert die Junktorenlogik, wenn sie mehrwertig erweitert wird, die Logik um Begriffe, die der klassischen Logik prinzipiell unzugänglich sind.

So verstanden und in mehrwertigen Wertetafeln formalisiert, ist es nicht eine unverbrüchliche Wahrheit, die sichtbar wird. Sondern die Strittigkeit und Unsicherheit verschiedener Erklärungen wird deutlich und operationalisierbar.
Während in der zweiwertigen Logik Strittigkeit und damit das Auftauchen eines urteilenden Subjekts schon rein formal garnicht vorgesehen war.

Außerdem zeigt sich, was sich aber auch schon in der klassischen Logik zeigt: Nämlich, ob die internen logischen Regeln korrekt angewendet worden sind. Weil die exakte Unterscheidung von Bedeutung und Form evident wird.

Der Philosoph und Mathematiker Leibniz hing seinerzeit der Illusion an, alle Entscheidungsfragen, alle logischen Schlüsse, ließen sich letztlich zu Rechenaufgaben machen.
Sein Ausruf "Rechnen wir`s" mit dem Absolutheitsanspruch der aus Descartes`s "Mathesis Universalis" von ihm in seine "Charakteristica Universalis" weitergedachte, formalisierte Wissenschaftssprache, die er aufzustellen gedachte, wird, als Resultat der hier angesprochenen Überlegungen, zurückgestutzt auf ihre tatsächlich mögliche Einlösung:

Noch vor allem formal korrekten Rechnen liegen Einschätzungen, in denen Beobachtungsgabe, Erfahrung, seelische, körperliche und aktuelle Befindlichkeiten und unbenanntes Denken ineinander fließen. Erst dann, wenn man sich entschieden hat, eine Beobachtung gemäß eigener Ansichten zu deuten oder eine Voraussetzung gemäß eigener Einschätzung zu machen, wird man diese, wie hier erörtert, formalisieren können.

Man sollte also, auch ganz generell, die menschliche Fähigkeit zur Selbstreflexion nutzen: die Fähigkeit, sich als ein erfahrendes und erkennendes und somit sich entwickelndes Selbst zu erleben: Wie man seine Beobachtungen macht und wie man zum Urteilen kommt.

Denn wenn die Einschätzung falsch war, wird auch korrekte Formalisierung an den Tatsachen nichts ändern. Akkurates Rechnen, aber auf falscher Datenbasis, bleibt falsch.
Es bleiben dies bessere oder schlechtere, jedenfalls von Menschen zum Ausdruck gebrachte, Erklärungen in und von einer überkomplexen Welt.