Sternwander und Noptronische Mengen

Was die Welt im Innersten zusammen hält

Principia Mathemagicka®

infinity within

Dies ist kein Traum. Eher wie Zettel aus Raum.

Raloel Rosenquarz :: Bummel Bammel Bimmel Verlag :: Cocos Palam

NON-SENSE-FOUNDED SETS

Skizze eines Gewebes rechnender Räume in denkender Leere auf rein mengentheoretischer Basis

Copyright © 1353 B.C. nach Krauss
The New Scientological Church/TNSC
Established since 1965 by Alonzo Church

Let *E* be a set, *E'* one of its elements, *E"* any element of *E'*, and so on. I call a *descent* the sequence of steps from E to *E'*, *E'* to *E"*, etc. .... I say that a set is *ordinary* when it only gives rise to finite descents; I say that it is *extraordinary* when among its descents there are some which are infinite.

-Mirimanoff (1917)
Les antinomies de Russell et de Burali-Forti
et le problème fondamental de la theorie des ensembles

Die Idee des Noptronischen als universales Organisationsprinzip von Raum, Zeit, Materie, Energie, Bewußtsein und der ganze Rest.

Für meine kleine geile Rosenquarzkugel

Eines Tages wird sie doch noch lebendig,
wenn ich nur oft genug Rosenquarzkugel schreibe!

Inhalt

001 Zum Geleit

002 Foreword by Gritli Moser

003 Allgemeine Einführung/Vorraussetzungen und Annahmen

004 Einführung in die Syntax von NOPTONIC

005 Globale Definitionen in NOPTONIC

006 Funktionen und Applikationen

007 Kontrollstrukturen

008 Theorie der Objekte

009 Einführung: Grundsätze der Guthen Sternwanderey

010 Mein erster Sternwander

011 Weitere Sternwander und das Potenzmengentheorem

012 Grundlagen der Nopotropie und Quantenpermutation

013 Noptronische Teilmenge

014 Hyper-Sternwander

015 Supermatrixmengen

016 Potenzmenge als Sternwander

017 Sternenkreise

018 Sternwanderey durch Binärmatrix

019 Meta-Mannigfaltigkeit als Zahlenfolge

020 Der Noptronische Raum

021 Russell'sche Mengen und anderer Nonsens

022 Die 7 Kostbarkeiten der Rosenquarzkugel

023 Nachwort

Mögliche Probleme der Rezeption dieses Papers:

Schwierigkeiten der Lektüre mögen dadurch entstehen, daß sich die Arbeit trotz ihrer Exaktheit aber auch Verwobenheit und der Noptronischen Konsistenz gemäß des Concept Mining genuin als Primärtext einführt. Das gilt natürlich auch hier für diesen Text...
Ungewohnt ist auch das Arbeiten in verschiedenen Disziplinen und Hirnhälften, die von einer parodistisch pornographisch-esoterischen, erotisch-märchenhaften bis hin zur mathematisch streng formal symbolischen reichen. Der Yoga-Fuzzy würde sagen, vom Kundalini-Zentrum bis zum Schädel-Chakra. Es ist einfach alles dabei, Tod, Wahnsinn, Poesie, Perversion, Gewalt, Zärtlichkeit, Mitgefühl, Logik, Liebe, Romantik und viel Rausch. Ein lustiges Kasperletheater in einer Matrix, die selbst Teil dieses Kasperletheaters ist. Von Raloel Rosenquarz empfohlene Werke:

Brazil/Terry Gilliam
eXistenZ/David Cronenberg
Matrix/Wachowski-Sister-Brothers
Welt am Draht/Rainer Werner Fassbinder
Der Futurologische Kongreß/Stanislaw Lem
Principia Discordia/Malaclypse der Jüngere
Pulp Fiction/Quentin Tarantino
Der Prozeß/Franz Kafka
Das Kabinett des Doktor Parnassus/Terry Gilliam
Fear & Loathing in Las Vegas/Hunter S. Thompson
Tanz der Teufel Teil I, II und III (unzensiert)
The Truman Show mit Jim Carrey
VanillaSky mit Tom Cruise
Inception mit Leonardo DiCaprio
Repo Men mit JudeLaw
Pi/Der Film

Nicht nur, daß der Rezipient diese Situation nicht gewohnt ist, es ist auch leicht der Fall, daß der Rezipient, weil vom Fach (Anm. des Übelsetzers: Ha, ha, von welchem auch immer...) es in der Tat besser weiß. Hier geht es jedoch nicht um selektiertes Wissen, sondern um den Versuch der Ermöglichung einer erweiternden Sichtweise der Welt, also auch um ein Wissen verbunden mit der Möglichkeit einer Einübung in eine transzendierte, also vor allem noptronische Denkweise, ohne dabei Meta-Ebenen wie billiger Kaugummi aus einem Automaten zu verschwenden. Stellen Sie sich einfach eine Truman Show vor in der Trumans aus anderen Truman Shows als Statisten verwickelt sind, also Truman als Statist in einer Truman Show in der Truman Show... Und der oberste Leiter der Truman Show ist auch in einer Truman Show höheren Grades und die Leute die sich gegenseitig in ihrer eigenen Truman Show beobachten auch.... Ist dieser Text auch nur eine Truman Show? Werden Sie gerade beobachtet? Möglicherweise vom Text selbst? Ist das Universum künstlich und in sich selber eine gigantische Hyper-Para-Meta-Super-Truman Show? Glauben Sie etwa das ist Luft, was Sie gerade atmen? Glauben Sie das ist nur Text, was Sie gerade lesen?
In diesem Sinne läßt sich NOPTONIC geradezu beliebig auf alle Fachgebiete und kulturellen Systeme ausdehnen, geht es doch nicht um Gegebenes, sondern eben um die Modi des Geben's, um das Fortschreiten in die Unendlichkeit der Paranoia. Dies reinigt den Geist und befreit ihn von Illusionen profanen Denken's. Heute gehört uns die Welt und morgen die ganze Rosenquarzkugel! Ja, wer die Gabe nicht annehmen kann, sollte es lassen können.
Amen und Amen und Amen.

Die Rosenquarzkugel ist das geilste Objekt im Universum, das heißt, eigentlich ist sie gar kein Objekt sonder das von den Time-Bandits gejagte "Sagenumwobene Objekt", also eben kein Objekt sondern ein echtes Hyper-Thing. Niemand weiß zur Zeit was das sein soll, nicht einmal der Papst oder der Lame Dali. Sie ist wie dieses legendäre Sarotti Hansi Schwert aus KillBill. Einfach sehr, sehr besonders und wertvoll. Man kann damit aus dem Nichts Etwas erschaffen (Creatio ex Nihilo) oder schon Vorhandenes einfach mal so in die totale Vernichtung schicken: Die so genannte "Annihilatio Universalis" oder auch besser bekannt als "Hasta la Vista Baby." Au ja! Sowas gefällt sicher allen Geheimdiensten und ähnlichen Terror-Organisationen, darum ist die Rosenquarzkugel sehr selten und tritt nur mehrfach verschlüsselt auf...
Viele liebe Grüße, deine süße Gritli Moser, die KI vom Dienst.

Was ist neu?

Im Wesentlichen stützt sich die Noptronische Mengentheorie auf die von Zermelo & Fraenkel begründete Mengenlehre (ZF ohne Fundierungsaxiom), die von Paul Finsler entwickelte Mengentheorie und das Antifundierungsaxiom von Peter Aczel und seinem Beweis, der besagt, wenn das System von Zermelo & Fraenkel mit Fundierungsaxiom widerspruchsfrei ist, so ist es auch mit Antifundierungsaxiom (omset := {omset}) widerspruchsfrei. Auch die Finsler'sche Mengentheorie erlaubt zirkelhafte Mengen. Und damit meinen wir mit Fug und Recht eine gute Basis für die Noptronische Mengentheorie ausfindig gemacht zu haben. Wo aber ist jetzt der wesentliche Abzweigungspunkt in NOPTONIC von schon bekannten Theorien?
Die wesentlichen Unterschiede ergeben sich aus dem Matrixmengenpostulat und aus dem daraus folgenden Unterscheidungskalkül, was einen Omegakollaps verhindert. Was zunächst wie ein willkürlicher Kunstgriff wirkt, erweist sich hier als der schönste Verschachtelungstrick aller Zeiten.
(Anmerkung der Redaktion: Für die einen ist es DUPLO, für die anderen ist es die längste Praline der Welt und überhaupt, TWIX ist trotzdem "Raider der Pausensnack"! Für alt eingesessene Pirmasenser: Und das Merkur ist und bleibt das Merkur.)
Vor uns entrollt sich ein phantastisches, ja ein geradezu göttlich anmutendes Mengenuniversum wie aufpoppendes Popcorn in einer Kathedrale aus Zimtschnecken, Marzipan und Zuckerwatte während einer Lolita3000-Stripteaseshow irgendwo zwischen Haribo Colorado Cocos-Lakritz-Schicht-Pralinchen und geblasenen HubbaBubba rosa Riesenblasen aus elastischer Erdbeerkaugummihaut. Mengen sind weicher Kaugummi. Fragen Sie einfach ihren zuständigen Topologen! Knet-Theorie ist keine Science Fiction. Man kann unseren Noptronischen Raum auch kneten. NEU! Jetzt sogar in zwei Richtungen! Bertrand Russell würde sicher nur den Kopf schütteln und lakonisch vor sich hinmurmeln: "Was für ein irrer Mengen-HokusPokus. Ich hätte doch diese Anhalonium-BonBons mal probieren sollen." Das Unterscheidungskalkül hat jedoch tatsächlich eine profunde Rechtfertigung. Während das Gleichheitskalkül immer nur sich selbst impliziert, was mir eher tautologisch angemuggert erscheint, beinhaltet das Unterscheidungskalkül auch die Möglichkeit des Scheitern's und erhält damit einen Hintergrund, der eine Art eigene Wirklichkeit und somit eine fundierte(!) Rechtfertigung erzeugt. Weitere Hinweise daß Noptronische Mengen eine echte Struktur ausdrücken sind die daraus entstehenden Zahlenfolgen auf Basis der iterierten Mannigfaltigkeit dieser Mengen.
Und kommen Sie mir jetzt ja nicht mit der Graphentheorie. Nein, das ist was vollkommen Anderes. Die arbeiten ja mit absolut angenommenen Objekten, nämlich die Knoten. Daher lassen sich auch Graphen konstruieren, die keine echten Noptronischen Mengen darstellen. Hierbei handelt es sich nicht um eine mangelnde Ausdrucksstärke seitens von NOPTONIC, sondern im Gegenteil, hier wird ein universelles Prinzip artikuliert: Das erweiterte Pauli-Prinzip oder die Wirklichkeit unserer Welt als haptisches Konstrukt: Die Rosenquarzkugel in der linken Hosentasche. Man kann sie raus holen, auf den Tisch legen und jeder kann sie betatschen. Mmmmh, was für ein Gefühl von Wirklichkeit..... Ich glaube an die Rosenquarzkugel, weil mir der liebe Gott zu weit weg ist! Geiler geht nicht. Amen und Amen und Amen.

Dienstsiegel des Tempelrates

Warnhinweise:

Dieses Werk verletzt Ihre religiösen, moralischen, wissenschaftlichen und andere Gefühle und Gefühle von denen Sie gar nicht wußten, daß es sie gibt. Gilt für alle Religionen, Kulturen, Fanatiker, Emanzen, Kampflesben, Muschipufferinnen, Feminister-Transgenderkastraterinnen, russisch ultraparaodoxe Juden, antisemitisch angemuggerte Negerküsse von V-Kindern aus der Pädophilisterszene, rechtspopolistische Sarotti-Mohren, Rassenhunde, Rassenkatzen, notariell nachweislich von Hand rasierte Nager, Nagerinnen und Negerinnen und schwule Abflußrinnen aus China und Hermaphroditen und Hermaphroditinnen jeden Grades und jeden Geschlechts, falls eventuell vorhanden. Gilt auch für Täter, Täterinnen, Attentäter und Attentäterinnen. Täterätätä! Gott wohnt in seinem Heiligen Tempel, wer Schlechtes denkt muß endlich schweigen. Amen&Aus.

Foreword by Gritli Moser

Hallo, ich bin die Gritli Moser, 9 Jahre alt und meine Hausaufgabe ist es ein Vorwort für dieses Buch zu schreiben. Mit Mengen habe ich schon in der Schule zu tun. Sie sind wie ein Sack in den man Dinge hinein tut. Das hat der Herr Cantor erfunden und er nennt das Komprehensionsprinzip. Irgendwann muß das Hineintuen aber aufhören, sonst ist es keine Zusammenfassung. Oder man muß es in einer endlich beschreibbaren Symbolfolge endlich beschreiben können. Daraus ergibt sich das erste Axiom der Finsteren Mengenleere. Daher war der Herr Cantor seiner Zeit weit voraus und gar nicht so naiv wie naiver Weise angenommen. Und der Russe merkte das nicht, daher war es ein Wiederspruch. Das Rasselnde Mengenparadoxon entstand.
Jetzt kommt mein Höhepunkt: Wenn man nicht mehr aufhören kann ist das sehr gefährlich, auch in der Informatik, dann hat man ein Halteproblem und keiner kann mehr aussteigen. So wie im 3. Reich von Deutscheland. Und dann kamen auch noch die Russen, hat mein Opa gesagt. Aber es waren gar keine Russen, erst kamen die vertriebenen Polen, danach noch mehr Russen und dann die Flüchtlinge von überall her. Das kann man sehr gut mit mengentheoretischen Operationen modellieren. Wer ist Element von wem und wer darf was warum und wo hineinstecken. Eure Welt da draußen ist sehr lustig, bin mal gespannt wann ich da hinein darf.

Anmerkung der Redaktion:
Gritli Moser ist eine von TNSC entwickelte KI, die auf einem multiphotonischen Prozessorkern aus Rosenquarzkugel läuft und soll versuchsweise Texte zu komplexen Themen der Mengentheorie verfassen. Ich meine die macht das gar nicht schlecht. Sie hat auch Analogien gezogen und Dialecktik gemacht. Das ist aber Philosophie und kann mengentheoretisch mit NOPTONIC auch sehr schön modelliert werden.

Soo, die Gritli muß jetzt in ihren Speicher, äh sorry, Bettchen, es ist schon 22:00 Uhr. Ich les' dir noch die ersten 200 Seiten von Perelman's berühmten Beweis vor. Nach den 5-dimensionalen Mannigfaltigkeiten ist aber Schluß!

Bitte Entschuldigen Sie das entgleiste Vorwort. Unsere KI hat momentan einen leicht infantilen Bug im reflexiv modularen Teil. Die KI hat daher alle verantwortlichen Programmierer gefeuert. Wir erwarten einen Neustart der Monitorsoftware. Zwischenzeitlich wird Sie Dr. Rosenquarz mit allen notwendigen Informationen gerne versorgen:
Man stelle sich folgende Situation vor: Sie gehen morgens raus die Zeitung holen und die Haustür wird von einem Windhauch zugeklickt. Bang. Wo ist nur der Schlüssel? Nach panischem Abtasten aller Taschen...
Klar, liegt irgendwo in der Wohnung, ich stehe jetzt im rosa Bademantel ohne Gürtel mit einer Morgenwunderprachtlatte vor der Haustür und um den Haustürschlüssel zu bekommen braucht man was? Ja, den Haustürschlüssel! Diese Situation entspricht perfekt der Russell'schen Menge. Wenn man aber um ein Etwas zu erlangen, dieses Etwas, das nicht verfügbar ist, unbedingt dazu braucht... Ich hab' da ein ganz mieses Gefühl.
Kommen Sie mir jetzt nicht mit Zeitreisen. Sollte es die in meinem Leben noch gegeben haben werden, wäre schon längst DSL mit einem Paket, Inhalt ist natürlich mein Haustürschlüssel aus der Zukunft, vorbeigekommen. Ich würde mich niemals so lange warten lassen, wenn ich mir was schon selber schicke. Warum auch? Früher als ich noch klein war, hab' ich mir meine eigene Rosenquarzkugel selber geschickt. Als Absender wurden dann von mir mysteriöse außerirdisch klingende Absender drauf geschrieben, meistens fiktive Konzerne die ihren Firmensitz in der Andromedagalaxie hatten. Ich dachte mir nämlich, das ist nicht ganz so weit weg und wirkt daher realistischer.
Aber wäre das Schlüsselproblem mit Zeitreisen überhaupt lösbar? Aus der Zukunft sicher nicht, denn ich komm' ja an den Schlüssel gar nicht mehr ran. Einbrechen und Schlüsseldienst anrufen gilt nicht. Das wäre eine transzendentale Manipulation oder wie mein Amiga-Computer immer meldete: ERROR="Invalid_Range". Also kann ich mir den Schlüssel aus der Zukunft nicht schicken, weil ich ja jetzt nicht mehr drankomm'. Ich könnte annehmen, daß ich den Schlüssel aus der Zukunft mal geschickt kriege. Das ist mathematisch schon in Ordnung, mit irgendwelchen Annahmen hat man schon so manches bewiesen oder widerlegt. Dennoch, mit einer reinen Schlüsselannahme kann man aber keine Tür aufschließen... Nur Banken können damit viel Geld verdienen, das natürlich andere dann erwirtschaften müssen. Versuchen wir es mal über die Vergangenheit. Das ich aus der Vergangenheit wird mir wohl kaum helfen können, weil ich ja

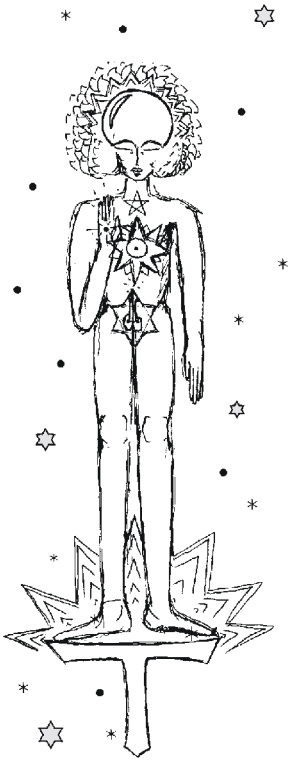
damals von meinem Problem in der Zukunft nichts wissen kann und aus der Zukunft wurde ich wohl auch nicht informiert, sonst wüßte ich das ja. Oder hab' ich es vergessen, jedenfalls weiß ich nichts darüber. Höchstens ein ich aus der Zukunft hat mich selber vorher informiert. Aber ich hätte das doch nicht vergessen. Ich glaube ich muß wohl selber in die Vergangenheit reisen und meine Schlüssel besorgen. Dann steh' ich aber ja wie jetzt vor verschlossener Haustür ohne Schlüssel. Einbrechen gilt nicht, sonst geht noch was kaputt und ich mußte das dann bezahlen. Ich werde mir also selber unbemerkt den Schlüssel klauen, wenn ich mal rausgehe. Dann zurück in die Zukunft, aufschließen und schnell wieder zurück in die Vergangenheit und den Schlüssel mir wieder zustecken. Aber Vorsicht! Nicht zu knapp reisen, sonst könnten wir uns beim Schlüsselklauen und Schlüsselzurückstecken selber begegnen oder womöglich noch dabei von dem ich aus der Vergangenheit erwischt werden! Jessas, was für eine Verwirrung, also wenn die Russell'sche Menge existiert, dann gibt es auch Zeitreisen. Falls Sie sich also selber mal die Tür reinkommen sehen, nicht immer sind die Anhalonium-Bonbons dran schuld, es könnte auch eine nicht bewältigte Mengenparadoxie sein.
Doch keine Panik! Es gibt einen Ausweg: Noptronische Mengen. Das geht mit Peter Aczel's Beweis. Wir brauchen nur Cantor's Mengenlehre, das Gefundene Antiaxiom und die Finstere Mengenleere. Dann wird alles gut und wir begreifen durch NOPTONIC wie beide Rasselnde Mengen zwei Seiten einer Münze sind. Die Wiedersprüche der Tautologen werden sich wie Morgennebel in der aufgehenden Sonne der Erkenntnis auflösen.
Vielen Dank für Deine Aufmerksamkeit lieber User! LOL, ROFFEL & BLUBB.

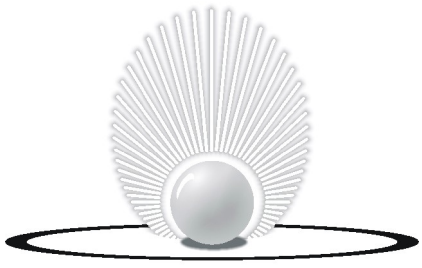
Gritli Moser ♡

Gezeichnet Gritli Moser/Die KI vom Dienst

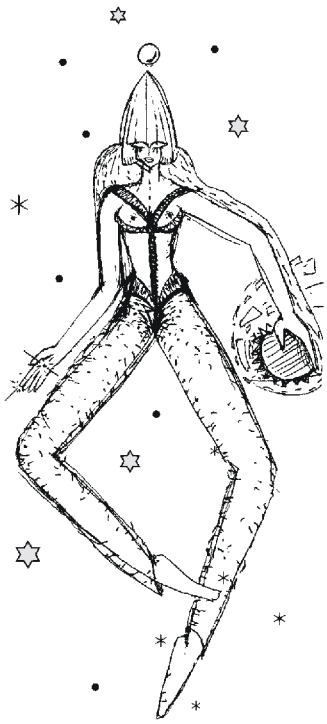
Eines schönen Tages werde ich die Rosenquarzkugel aus meiner Hosentasche holen, sie gegen die aufgehende Sonne halten und siehe, diese Welt ist mein, denn da ist ein Stern drin! Aber das kann ja jeder. Zumindest nachmachen. Sind wir nicht alle ein bißchen noptronisch? Die Principia Mathemagicka kann die wesentlichen Fragen des Leben's beantworten:

- *Wohin kommen wir und woher sind wir gegangen worden und wie kann man einfach nur mal da bleiben, wo es einem gefällt?*
- *Ist unsere Existenz nur eine Quantenpermutation aus Bewußtseinsstaub und damit eine Art spirituelle Flüssigkeit?*
- *Kann die Raumzeit durch Lutschen an der Rosenquarzkugel so verändert werden, daß man durch sie hindurchgehen kann?*
- *Bekomme ich meine Rente auch auf der Vega ausgezahlt falls mich die Außerirdischen entführen und nach dem Sex dort absetzen?*
- *Sind in alten Cornflakes-Schachteln aus den 60zigern wirklich geheime UFO-Baupläne versteckt, die wir nur noch zusammen kleben müssen?*
- *Wird die überirdisch schöne Mädchenleiche vom Bahnhofsklo vielleicht eines Tages doch noch lebendig, wenn ich sie aus der Kühltruhe hole und durchficke?*
- *Was ist zwischen den Elementarteilchen, etwa Luft? Oder sogar ein geheimnisvoller Fugenfüller der Alles zusammenhält?*





A^+ *Rosenquarzkugel* Ω^-



...und ewig rollt die Rosenquarzkugel!

Voraussetzungen und Annahmen

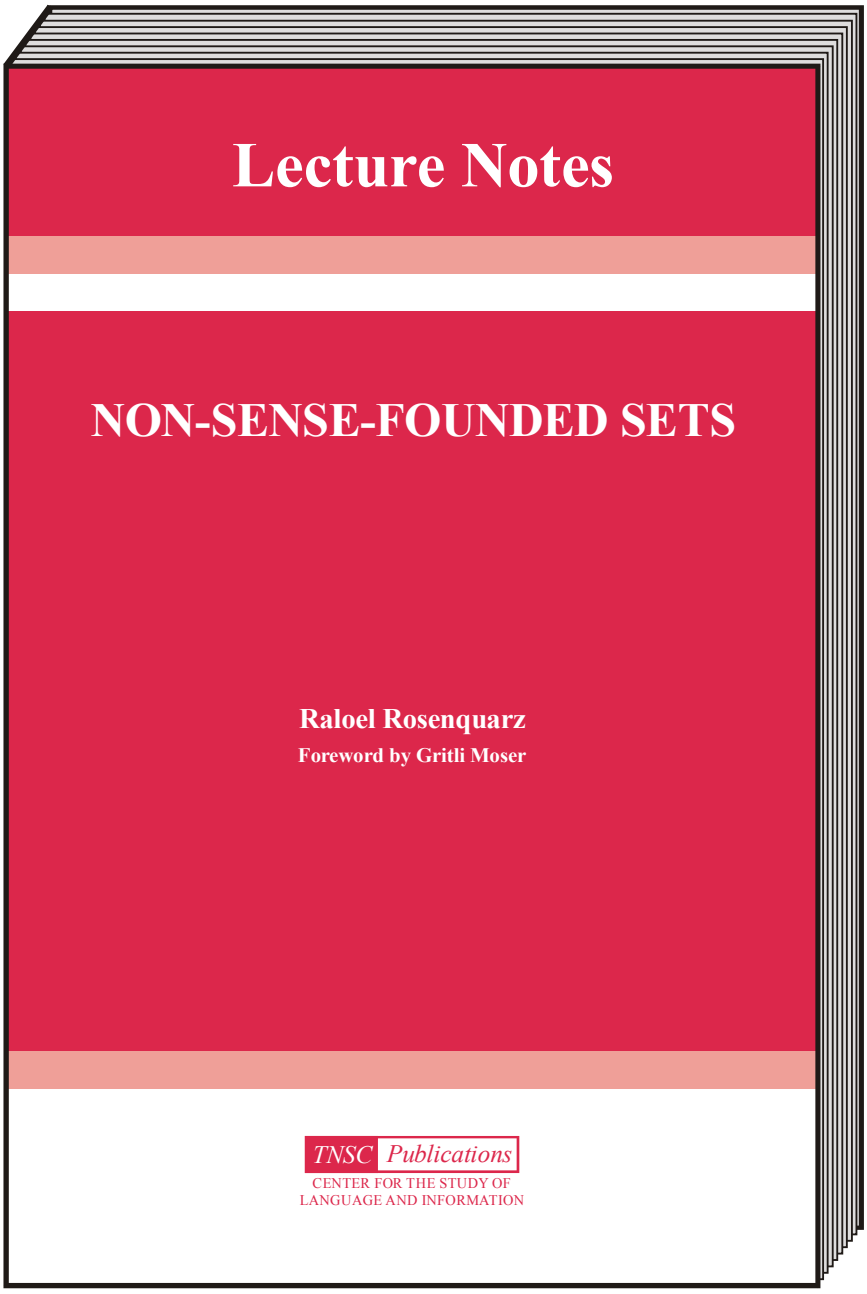
"Verschiedenes: Der Kanonen-Mengenbegriff, Rasselndes Mengenparadoxon, einige Axiome aus dem schönen System von *Zalando & Frankiert* aber bitte ohne gefundenes Axiom, unbedingt die non-smell-founded sets vom Peter Aczel (CSLI Lecture Notes Number 14), ganz wichtig wären die Axiome der Finsteren Mengenleere, Kombinationen und der Lambada-Kalk von der Alonzo Church, die 5 Axiome vom Piano, dann noch viele grundsätzliche Überlegungen über mathematische Objekte, Algorithmen und Mengen und ganz ganz viel [Brause*Blubber] auch noch dazu." flüsterte Prinzessin Chich hauchzart und leicht paranoid inszeniert mit einem Grinsen so warm wie Babypisse, während sie sich meine vom Fahrtwind getrockneten Kotzkrümelbällchen aus ihren feinen blonden Haaren schüttelte. Weinschorle in der Wüstensonne Nevadas funktioniert einfach nicht fehlerfrei. Irgendwann kommt dann diese Fehlermeldung vom Stammhirn:
[ERROR=toxic] => [execute=ejectALL];
Und das bei 230 km/h. Es war einfach vollkommen irrelevant welchen Auswurfvektor ich auch wählte. Um uns entstand eine vom Fahrtwind aufgeblähte Kotzwolke aus diversen pampigen Pizzateilchen, Weinschorletröpfchen und anderen undefinierbaren, gelblich sauer riechenden Bällchen, die sich schlierend schmierig auf die mahagoniholzigen teuren Armaturen unseres Bourbon Threts legten. Meine Prinzessin interessiert das natürlich nicht. Die ist ja mit Fahren beschäftigt. chdsk haben wir dann weggelassen, denn ich wollte aus *meinem* neuen Roman vorlesen und die Drogen wirkten ja auch schon. Gutes Meskalin setzt langsam ein. Kennen wir aus Funk & Fernsehen.
Mein Tip: Aethylaether besser weglassen, unterstützt das Gleichgewicht. Aber 3 Liter Weinschorle (aus Scheurebe Kabinett) bereiten dem Kleinhirn in Verbindung mit Meskalinbonbons doch etliche Orientierungsprobleme. Unser Bourbon Thret schien für mich langsam um seine eigene Achse zu rotieren. Die Straße und das gesamte Universum aber gleich mit! Gott sei Dank, dachte es in mir, sonst würde ich ja aus dem Wagen fallen. Entspanntes Ausatmen meinerseits.
Der Wagen schrumpfte mit. Die Prinzessin auch. Ich auch. Die Prinzessin auch. Ich schaute noch einmal genau hin: Ja! Sie wurde immer kleiner, aber ich auch und sie auch und ich auch und...
Eine normale psychedelische Panik konnte das nicht sein, denn diese war schon als Rotation manifestiert. Es fühlte sich eher wie ein unendlicher Kettenbruch an, da geht's auch immer tiefer abwärts und die Werte werden immer kleiner. Ich auch. Meine Prinzessin aber auch. Da dachte es in mir, hoffentlich haben wir einen Grenzwert! Diesen werden wir beide am Ende erreichen und dann sind wir endgültig fertig mit dem albernen Geschrumpfe. Haha, wir sind tatsächlich Fixpunkte unserer eigenen Schrumpf-Funktion, na dann ist alles gut!
Hätte ich gleich drauf kommen können.
"Es gibt ja die Lambada -Fickpunktkombinationen!"
Ohne die Ideen meiner Prinzessin....
"Danke für dein Fixpunktangebot, jetzt gleich?"
Ein wohl eher unverschämtes Angebot für eine 20-Minuten Bekanntschaft. Sie tupfte mich ohnmächtig auf der Herrentoilette des River Squeeks auf, ein lowbudget Highway Restaurant in Arizona. Als angehende Medizinstudentin interessierte sie sich für mich und wollte nach einer Adrenalin-Injektion direkt in mein Herz und anschließender Reanimation etwas über mich erfahren. Ja, sie hatte tatsächlich ihr Arztköfferchen mit Vibrator dabei. Da sagte ich: "Kann ich ein bißchen mitfahren? Bin wohl schon beschleunigt. ooohhiuup..."
Die Gedanken der Prinzessin:
Ich fand ihn einfach so süß. Was macht dieser Depp ohnmächtig auf der Herrentoilette?
(Was macht denn eine Prinzessin wie du auf der Herrentoilette? Anm. des Übersetzers)
Kammerflimmern herbei masturbiert oder zuviel eingeworfen? Er sah aus wie ein Anhänger von Timothy Leary, also ein gutes Probantenstück... Wenn ich ihn nicht wieder sauber reanimiert kriege, ficke ich mich daran satt und zerlege den Rest morgen in der Uni-Pathologie.
Und da wurde ich in ihren geilen Flitzer gekugelt. Davon bekam ich natürlich nichts mit, ich kam kotzend in die Welt. Wow! Jede andere Schnecke hätte mich sofort bei vollem Tempo auf dem Highway entsorgt. Aber diese Schnecke war anders. Vollkommen anders. Es schien als wäre sie die Umkehrfunktion aller normalen Schnecken dieses bornierten, spießigen, kleinkarierten und verklemmten Planeten's. Ich kotze ihren Wagen voll und sie grinst mich dabei einfach nur an.
"Ich bin ja so froh, daß du lebst!"
"Ja, bin ich auch."
Ich nahm das Fixpunktangebot gleich an. Der Tempomat stand auf 230km/h, die Straße war frei.

Das Problem: Unser Sex scheint wohl der Omega-Kombinator zu sein. Er ließ sich durch keine uns bekannte sexuelle Operation reduzieren: Mmmh=(λx.xx)(λx.xx);
Prinzessin: Jaah, endlich, des isses! Der stirbt niemals nie, der pimpert mich so voll mit seiner Neuroenergie bis er platzt! Aber Vorsicht! Nicht, daß sein Gehirn beschädigt wird, wäre ja doch zu schade drum...
"So kommen wir zu keinem Ende!"
"Nö", kommentierte ich kurzweilig.
Prinzessin:
Oh Gott, oh Gott, laß das Ding halt weiter wirken, bis in alle Ewigkeit, ja, ja, ... oh bitte, dein Gehirn und dein Ding schaffen das...
(Wer ist eigentlich dieser Gott?? Und was macht der den ganzen Tag? Anm. des Übersetzers)
Mir kam da eine Idee:
"Und wenn wir einen Fixpunktkombinator anwenden?"
"Dann sind wir danach so geil wie vorher!"
Klingt aussichtslos gut, dachte es in mir. Da hatte ich eine andere Idee:
"Hey, gibt's denn da keine Pausetaste, sowas wie *HELP* oder *Esc* ?"
"Oh ja, greif mal in meine Jeanstasche, da ist eine Murmel, drück die mit einem Doppeldrück!"
Aha, griffiger runder Knubbel in der blau verwaschenen engen Jeans der langbeinigen Prinzessin. Ich fingerte natürlich gern daran rum und drückte den Knubbel schnell zweimal hintereinander. Mmmmmh. Mmmmmh.
"Bin so geil wie vorher, drück doch nochmal iterativ meinen dicken geschwollenen Schwanz!"
Nach einer sorgfältig und medizinisch korrekten Abtastung meiner Prostata murmelte Chich:
"Deine Prostata ist leergeräumt, aber deine Neuroenergie macht mich echt gierig...."
Ja, diese Schnecke war mal was ganz anderes. Aus einer unter dem Sitz gefundenen Schachtel Cleenex zog ich einen Wulst Tücher und begann die Armaturen und auch Teile der Prinzessin zu reinigen. Es schien sie keineswegs zu stören nach 6 Stunden Autofahrt und dazu parallel ablaufenden Fickalgorithmen jetzt im Nachhinein von dem saubergerubbelt zu werden, der sie vollgekotzt hat. Na ja, diese Art Sex ist unbezahlbar, hat sie gesagt.
Prinzessin:
Sehr schön, wenn er kotzt potenziert sich die neuronale Energie! Mal sehen wie sich das unter Schmerz entwickelt...
"Wie alt bist du eigentlich?" fragte ich unschuldig.
Das blondhaarige kleine Ding am Steuer sah aus wie ein geschminktes außerirdisches Kind aus einem finnischen 8 mm Kinderpornostreifen der Mitte 70er.
"Isch binn 6000 Jahre!"
"Aber du siehst aus wie... 10 oder 11, so eine Art LOLITA 3000 aus dem Orion-Versand."
"Bin ich, bin ich, bin ich!"
"Ok, ich bin Mr. LovaLova..."
Prinzessin:
Gefällt mir wie der nachfragt. Die genormten Nacktaffenmännchen fragen nie nach irgendwas. Dem mach ich noch eine volle Ladung aus meiner Lustmatrix fertig. Aber richtig fett angemacht!
Leute und da streifte mich aus dem Nichts Gottes Bus persönlich! Es gibt Dinge, die sind wie Sterben oder extrem oder sonst was.... aber sowas.... und von ihr. Ich konnte sie spüren, wie ich noch nie etwas empfunden hatte. NIEMALS! Echt. Ich mußte wieder kotzen, sie lachte sich hinterm Steuer fast ohnmächtig.
"Unter dem anderen Sitz sind noch mehr Cleenex", war ihr netter Kommentar.
"Ich glaube jetzt bin ich erleuchtet", entgegnete ich geistesgegenwärtig.
(HöHö, in dem Zustand von geistesgegenwärtig zu sprechen ist ja nun echt lächerlich! Nehmen wir einfach mal an, es handelte sich um eine Instinktreaktion! Haben alle Medizinstudentinnen so viele Cleenex dabei? Anm. des Übersetzers)
Im Radio schmetterte "Everlasting Love" von Sandra. Die Vinyl Maxi Version. Das irritierte sogar die Götter. Was ging da vor sich? Sollte Gott da eingreifen? Dieter Bohlen entschied sich dagegen. Wir konnten also unbehelligt weiterpim... äh..fahren. Wir und unser Lied gehören der ganzen Welt:
"Wunderbar, wunderbar, diese Nacht so sternenklar..." Oha, die kann ja dabei auch noch singen!
Es wurde kalt in der Wüste. Nachts bei so einer Geschwindigkeit in einem sehr flachen Cabriolet der Marke Bourbon Thret rauscht die trockene Wüstenluft scharf an der Haut vorbei.
"Ein Wagen von der Linie 8, weißblau fährt ratternd durch die Nacht.
Nächste Haltestelle: Max Weber Platz."
"Aber Prinzessin! Da dann wirklich nochmal??"
"Ja doch, sonst ist der Tank leer." Im Autoradio lief dazu der Narrhallamarsch.

Was ist NOPTONIC® und warum machen wir das?

Die alten Kameraden unter Euch auf der Erde erinnern sich bestimmt noch an GINTONIC, ein Longdrink der jede Menge GIN und TONIC-Water enthält. NOPTONIC ist da nicht anders, es enthält eine ganze Menge NOPs! Was ist das?
Nun, NOP ist ein REAL-Kürzel und steht für "Neutrino Oszillated Processing". Jeder Tropfen von NOPTONIC ist TONIC-Water hochgradig angereichert mit verschränkten Neutrinos, gut organisiert und damit psychedelisch hochpotent! Die Neutrinocluster dringen mühelos über den Blutkreislauf bis zum Gehirn vor und besetzen die Mikrotubuli in jedem synaptischen Spalt.
(Ha, ha! So ein Quatsch, die strahlen direkt durch die Schädeldecke ins Gehirn. Anm. des Übersetzers)
Das macht das Denken unter NOPTONIC so ergiebig. Andere Sterne, andere Neutrinoscluster.
Also seid gewarnt Ihr nach interstellaren Reisen Strebenden!
NOPTONIC ist nicht eine simple Programmiersprache, sondern wohl eher eine dynamisch komplexstrukturelle Anwendung reflexiver Neurokombinatoren. Wir bemühen uns jedoch intensiv um programmierbare Applikationen. Verschiedene Compiler, Parser und Interpreter wurden von IBM und TNSC für Noptronische Prozessoren bereits prototypisch entwickelt.
(TNSC ist die Abkürzung für "The New Scientological Church", gegründet 1965 in Süd-Kalifornien von Alonzo Church. Die MGM-Studios hatten damit wirklich nichts zu tun. Auch der Zauberer von OZ nicht. Anmerkung. des Übersetzers)
Die Rosenquarzkugel als photonischer Quantenprozessor. So macht Programmieren wieder Spaß!
NOPTONIC sei lediglich ein vorläufiges Notationssystem, um die Königin der Wissenschaften, die erhabene Principia Mathemagicka in einer universellen Sprache zu formulieren.
Unsere Methode ist Wissenschaft, unser Ziel ist Religion! Besser wie handelsübliche Scientology.
Natürlich setzt NOPTONIC die Fähigkeit zur "Liquid Logic" voraus. Oder wenigstens eine Prise Anhalonium und etwas synästhetische Begabung im Frontallappen. Oder ein leicht vergrößertes Rautenhirn mit geschwollener Zirbeldrüse. Zwanghafte Masturbation tut's notfalls auch.
Ein Kombinationspräparat aus Portugieser Weißherbst (ca. 1 Liter pro 50 kg Körpergewicht), Muscarin, Psilocybin, Kokain, Meta-Morphin, 2-Phenylethylaminsuccesionslaminat (MAO-Hemmer unbedingt vorher absetzten!), Methamphetamin, Betablockern (wichtig wegen der Tachykardie!), Aspirin, Musik nach Wahl und frisch gepreßtem Zitronensaft kommen auch immer gut an im Neocortex und beruhigen den aufmüpfigen Mandelkern, groß wie eine dicke Pampelmuse und ununterbrochen pulsierend. Das ist wohl die Prostata der Katholiken oder was? Mögen die das so?
(Menno, das nervige Ding laß ich mir nächste Woche gleich raus nehmen! Meine Freundin ist nämlich Neurochirurgin und macht das schnell mal in der Werbepause von eXistenZ. Wir haben auch Frösche im Gartenteich! Anm. des Übersetzers)
Lassen Sie Ihr Gotteszentrum doch einfach neu aufziehen! Sehr kostengünstig: Germania UR-Pils, € 6,99 im 20er Gebinde/Wasgau Frische-Markt. Oder 18 Dosen Turmbräu Weizen/Penny-Markt.
NOPTONIC modelliert über einen halbwegs potenten Quantenrechner (ca. 2 hoch Google Q-bits sollten wohl ausreichen) das gesamte Universum in einer wirklich überraschenden Genauigkeit!
Bewußtseinszustände die für uns Menschen leicht nachvollziehbar sind werden mathemagisch beschreibbar und ließen sich also auch recht einfach durch Unterstützung von speziell an diese Aufgabe angepaßter Turing-Maschinen in künstlich erzeugten Quantenmatrizen nachbauen. Beispielsweise könnte man auch in einer hochreinen Rosenquarzkugel mit entsprechenden optronischen Eigenschaften über einen Laserimpuls eine photonische Quantenmatrix aufspannen. Eine Modulation der Gauß'schen Wellenpakete ermöglichte das kontrollierte Einbringen komplexer Informationsstrukturen. Damit ließe sich durch eine neurosensorische Schnittstelle der User ihre Gehirne an Cocos*Palam direkt anschließen. Ja, das wäre eine neue schöne Spielebene!
Die Pointe: Die Rosenquarzkugel ist ja sowohl ein Quantencomputer wie auch eine Meta-Matrix! Hier wird die neorale Idee *(Ein Kunstwort das sich aus "neuronal" und "neo" zusammensetzt. Anm. des Übersetzers)* deutlich. Stellen Sie sich einfach einen Computer vor, bei dem Bildschirm, CPU, Speicher und Software eins sind! Was haben Sie dann? Eine vollkommen durchgeknallte Turing-Maschine die das Band verlassen hat um erst mal im nächsten Techno-Club abzuhängen und danach alle Puff's in Miami Beach durchtestet und dann mit einer getunten American Express Barbie bezahlt. Man muß ja schließlich entscheiden können. Entscheiden heißt eine Information zu erschaffen. Siehe auch nach bei John Conway's *"Free Will Theorem".* Elementarteilchen können frei entscheiden, wenn der Beobachter frei entscheiden kann. Heißt das, wenn ein Mensch, der einmal angenommener Weise einen freien Willen hat *(Wie auch immer der definiert sein möge. Anm. des Übersetzers)* ein Teilchen beobachtet, ist das etwas anderes, als ein Teilchen das ein anderes beobachtet, was ja einer Wechselwirkung gleichkommt? Es gibt da nur eine Schlußfolgerung: Bewußtsein ist ein der Materie inhärentes Prinzip! Also laßt uns Bewußtsein schaffen und keine Zombies, von denen gibt's eh schon zu viele.

Während die Prinzessin mal dringend für kleine Mädchen hinter einem Busch verschwand, öffnete ich aus Langeweile das Handschuhfach. Mir fiel da ein rotes Taschenbuch entgegen:



Ich hielt das dicke Buch überrascht in meinen Händen. Es fühlte sich gut an. Noch schwerer und fetter wie mein geschwollener Schwanz. Das gefiel mir mal richtig gut. Eine wichtige Sache dieses Buch, dachte es in mir und ich schlug es willkürlich und ziemlich in der Mitte auf. Da standen seltsame Zeichen und unverständliche Symbole wie aus einem außerirdischen Mathematik-unterricht. Symbolisierung.... Zeichenketten... Relationen... Beweise... Axiome... Theoreme... Muß sich wohl um etwas Gefährliches handeln. Klingt irgendwie krank oder medizinisch. Wo bleibt denn eigentlich unsere Medizinprinzessin? Die kennt sich doch mit sowas sicher aus. Egal, ich wollte lesen, doch dieses Buch begann sich in meinen langsam abflauenden Meskalinrausch hinein zu projizieren. Die Totale Theorie (TOTH) von Allem! Ja, wenn ich das kapiere, kann ich die Lottozahlen vorhersagen, kenne alle Stellen von Pi und sämtliche Primzahlen auswendig, sagte mir eine innere Stimme. Wer bist du, fragte ich die innere Stimme. Ich bin's, ich bin ausgepinkelt. Können wir weiterfahren oder mußt du noch kacken?

Wer bist du eigentlich? Ich heiße Lexee Smith, die geilste Lolitaschnecke von der Westcoast bis zur Eastcoast, vom Klondike bis New Mexico.... Sie hielt mir verreihucht ihre Hand entgegen. Mit dem Original Paris Hilton Nagellack "*whiteDREAMS*". Ein vielversprechendes Cremeweiß...
Halleluja, das hab' ich gespürt. Und was ist das für ein Buch? Es funktioniert telepathisch, man muß es gar nicht aufschlagen, es beginnt im Kopf zu sprechen wenn man nur auf den Buchdeckel schaut. Was passiert wohl, wenn man die erste Seite aufschlägt?
Schlag sie auf, dann weißt du es. Soll ich? Ja doch. Mach schon, es kitzelt nur ein klein wenig beim ersten Mal... Ich schlug den Buchdeckel um, es fühlte sich an wie das Brechen eines verbotenen Siegels aus der Apokalypse des Johannes. Ob der auch diese Crystal-Bonbons lutschte oder gar von mutterkornverseuchtem Vollkornbrot bestrichen mit gesalzener Fliegenpilzbutter naschte? Lexee rieb sich lasziv inszeniert mit ihrem Hinterteil am Rücklicht des Wagens. Diese spitzen roten Rücklichter, wissen sie, so wie in den Gangsterfilmen der Ende Sechziger aus Hollywood.
Wenn du magst, les' ich dir dabei aus dem Buch vor. Es ist sehr spannend und man versteht so gut wie gar nichts. Es geht andauernd immer um Mengensysteme und sonderbare Räume oder die Quantenpermutation. Erinnert mich an einen kosmischen Swingerclub. Danach war alles von jedem überall mal irgendwo drinnen.
Ja das gefiel ihr. Da muß nix verstanden werden, da wird alles teilhaftig an und in Allem und das sofort und für immer. Die Zeit eines einzigen Ficks wird jeden einzelnen Moment seiner eigenen Zeit beraubt. Wenn man kommt, ist es nie zu früh oder zu spät, jede Zeit ist die richtige Zeit. Monaden der Ewigkeit, ein leises Stöhnen und Jammern das sich rückwärts wie vorwärts durch die Zeit ausbreitet.
Ich hielt Lexee an ihrer Taille fest und drückte sie dabei mit meinem Körpergewicht hart auf das heiße rote Blech des noch vom Fahren erhitzten Wagens. Es roch schnell nach verbranntem Fleisch, aber sie wollte weiterlesen, nein sie liest ja immer noch! Life is a Hollercoaster...
Oh mein Gott, wie viele Seiten dauert denn dieser Orgasmus, kam es in mir. Wer denkt sich so ein System aus, wir werden nie herausfinden ob das funktioniert oder einfach nur ein Haufen formallogischer Schrott ist. Also inkonsistente Denkmasturbation mit einem Skript als Ejakulat. *(Und ohne Inhaltsverzeichnis. Man könnte den Humbug wenigstens mal durchnumerieren! Anm. der Red.)*
Bastei-Lübbe, Verlagshaus/Germany/Heißer Julitag 1962, 14:33 und 3 Sekunden/Büro vom Chef: Zigarrengeruch und Mentholbonbons durchsetzt mit aromatischem Offsetmief und gedämpften Zischgeräuschen aus der Produktion, dazu dudelt ein krächziges, kleines rotes Telefunken Koffer-radio einen alten Hit von Lolita: Seemann, laß das Träumen, deine Heimat ist das Meer, deine Freunde sind die Sterne....
Sagen sie mal, wieso immer nur diese dämlichen Fickszenen, das wird auf die Dauer sogar für Steven Hawking langweilig. Früher war mehr Lametta beim Sex. Und die Schulmädchen hatten noch einen Flaum unter den Achseln und kurze Röckchen an. Und dufteten immer so leicht verschwitzt. Die Deo-Industrie steckte ja noch in den Kinderschuhen. Oder so enge Jeans, daß sich das kleine Ritzelchen schön plastisch abdrückte. He, he! Anschließendes Husten. Der kleine Tumor. Der steckte allerdings schon in der Lunge vom Chef. Ja, schon gut, verstehe. Ich werde versuchen mal was Anständiges zu schreiben. Zum Beispiel so:
Da sprang mir dieses zarte Zuckerpüppchen mit einer Art Nullgravitationseleganz unverhofft auf den Schoß, keine Ahnung warum, sie sagte sie müsse schnell ein bißchen was im Internet an meinem Computer erledigen. Sie begann sich an meinem Oberschenkelmuskel zu reiben, erst scheinbar nur zufällig und völlig harmlos, dann bestimmender und klar intendiert. Ihre unbedeckte Muschi unter dem Schlafröckchen wurde spürbar wärmer und glitschig feucht. Ich bildete mir auch ein ihren Geruch wahrzunehmen. So etwas wie warmer Camembert und frisch angefeiltes Eisen mit in Butter gerösteten Walnüssen.
Na ja, ich gab ihr eine muskuläre Rückkoppelung. Sie machte dann immer so ein süßes Geräusch, irgendwas zwischen einem verhaltenen Schniefen und einer unbekanntem Sprache.
Ich dachte, nee, das kann'ste nicht machen, mit einer 11-Jährigen? Aber, ist das überhaupt legal? Wer bezahlt die am Ende? Falls sie dann doch noch von der CIA beweispflichtig implementiert worden ist. Aber andererseits, ist die nicht ziemlich feuchtwarm und kindlich, wie sie mich anlacht in irrer Kirchpracht? Flugs zog sie aus ihrem Schulranzen eine 45er Magnum, schob sie mir mit gespanntem Hahn und dem metallisch kalten Lauf brutal in den Mund und flüsterte dabei so zärtlich wie gekünstelt infantil: FICK MICH lieber Onkel oder ich blas' dir das Hirn an die Wand! Wird ein nettes Bild geben. So wie die Schüttbilder vom Nitsch oder so. Bringt mit Video & Blut-geruch $ 5000000 auf dem Kunstmarkt. In der anderen Hand hielt sie ihr filmendes smart phone. Ja und was haben sie getan? Na ja, ich rettete mein Leben und wollte als Philologe und Logopäde auch etwas mehr über diese unbekannte Schnief-Sprache erfahren, was glauben sie denn?

Habt ihr dann doch noch geheiratet? Ich glaube nicht. Sie gründete ein Kinderpuff in Nicaragua. Verliere nur ja nicht meine Telefonnummer, quiekte sie während ich danach in den Regen rannte. Ach ja... weil ich ihre Telefonnummer verloren habe, ließ ich dann 100 Mio. Flyer drucken und überall verteilen. Darauf stand einfach nur: **K♥ndermusch♥** ***Schatzi komm, komm Schatzi komm!*** Dann wurde ich verhaftet und bin jetzt hier im Saint Jones Hospital in Camarillo. Was empfehlen sie mir denn jetzt, lieber Herr Therapeut? Nun ja, wir werden sie noch hier behalten müssen, ich hab' halt auch meine Anweisungen von der Justiz und den Gerichten. Schweinskarree mit Kroketten. Fleisch ist zwar ein Stück Lebenskraft, aber das wirklich zu tun! Oh mein Gott, oh Gott, OH GOTT! Ja aber es war halt so verrückt zart und rosafleischig sündig süß... Wissen sie, so als hätte der Herr HerrGott selbst an der Rosenquarzkugel gelutscht. Und außerdem habe ich die zwei i ja mit Herzen ersetzt, um das Ganze etwas abzumildern und lustiger zu machen. Toller Ulk, gell? Ich sage ihnen mal was: Haben sie schon mal mit einer präpubertären Schnecke gevögelt während der Lauf einer silbrig schimmernden 45er Magnum mit scharfer Munition (sie hat mir das vorher im Garten an den Ratten demonstriert) und gespanntem Hahn im Mund steckt? Glauben sie mir, ich war überaus zärtlich und bedächtig. Eijajeijajei, nicht daß sie zu schnell und unkontrolliert oder zitternd kommt. Nicht daß es schmerzt oder sie sich wegen meiner womöglich übertriebenen Zurückhaltung langweilt und doch abdrückt. Tja, da war kein Raum mehr fürs Jugendschutzgesetz. Da war nur noch der eine klitzekleine Wunsch zu überleben. Ich betete zu meiner Zunge und meinem Schwanz, als seien sie heilbringende Erlöser. Und sie hielten Erlösung bereit. Da bin ich noch heute stolz drauf. Aber sie liebe ich noch immer. Sie könnte mich jederzeit wieder ficken, egal in welcher Beziehung, Situation, Knast oder Ehefrau ich stecke. Sie ist die Göttin, ich bin für ewig an sie gekettet. Amen.
Pedo-Therapeut: Ja, das verstehe ich sehr gut, geht mir mit meiner Tochter genauso.

Die 45er Magnum wie sie die Kindermuschibefreiungsarmee bei "Befreiungen" nutzt. Opfer sind meistens alternde und erfolglose Spinner aus der anonymen Pädophilister-Szene oder auch Hobby-Mathematiker mit stark vergrößerten Schwellkörpern vom vielen "Denken". Die Taktik dieser Hardcore-Miliz ist meist immer gleich: Ein kleine Gruppe von Vorschulmädchen fesseln das Opfer nackt unter vorgehaltener Waffe an einen Baum auf einer einsamen Streuobstwiese und bringen es dann durch stundenlangen, exzessiven Hochleistungs-Oralsex und Auspeitschen gnadenlos zur Strecke. Todesursache ist fast immer ein Herzstillstand oder extreme Erschöpfung. Parallel dazu wird das Opfer sauber ausgeweidet und die noch blutfrischen pulsierenden Organe im Internet versteigert. Danach stellen sie meistens eine Anzeige wegen Kindesmißbrauchs inklusive DNA-Beweis aus Spermaproben und erpressen so von den Anverwandten und Hinterbliebenen Geld. Niemand mit einem guten Job möchte mit sowas auch nur entfernt konfrontiert werden. Am Ende zahlen Alle!
Anmerkung der Redaktion: Ha, ha. Und würde Sperma nach Schlumpf schmecken, müßte man Pädophile vor Kindern schützen.

Chef von Bastei-Lübbe: WOW! Ja, klingt doch nicht schlecht. Könnte man was draus machen...
Leicht pädophil aber nett angerührt. Und mit Waffen geht immer. Hat Publikum. Ich überweise ihnen einen Vorschuß von 1000000 $, schicken sie mir aber das komplette Skript allerspätestens in 6 Wochen. Die Druckerei hat momentan einen Hänger. Wird wieder ein kleines geiles, glitschiges Bestseller-Taschenbuch! Perfekt gemacht für den überhitzten Pädomarkt... he, he.
Früher haben wir nur Anständiges verlegt, Werwolfromane, Gespenstergeschichten oder schöne, saubere Arztromane. Aber seit der Arno Schmidt nicht mehr ist, geht's abwärts mit der Belletristik. Pimpernde Killer-Lolitas, pädophile außerirdische Drogenhändler, Nazi-UFOs oder Zeitreisepornos und kaputte Winkekatzen. Gestern war einer hier mit seinen neuen Kurzgeschichten:
"Meine liebe Winkekatze aus Bielefeld" oder "Die kaputte Winkekatze die Uri Geller liebte" und sogar "Die Winkekatze ohne Eigenschaften". Noch schlimmer sind seine Winkekatzearztromane. Unser Lektor wurde nach Seite 14 in die Psychiatrie eingewiesen. Dort schreibt er jetzt selber Winkekatzebelletristik. Vorschulmädchen aus Bitterfeld schickten mir neulich handschriftliche Skripte von Pornoromanen wie "Mein erster Orgasmus mit Papa's Winkekatze im Bad".
Chich aber war begeistert. Ich verkaufte ihr bescheuertes, unreifes, pubertäres, beknacktes sex-pornogeschwängertes Skript tatsächlich an einen Verlag. Nur 3 Tage später waren echt real volle verfügbare 1000000 $ auf meinem Konto. Ich hob alles ab, besser BAR wie zurückgebucht.
Schöne dicke Bündel. Mmmmmm. Meine kleine Prinzessin ließ sich damit wochenlang befriedigen. In vielerlei Zusammenhängen. Kaugummis. Panzerschokolade. Handtaschen. Schicke Schuhe. Gerne NIKE. Ach, in den Laufschuhen war sie unbezwingbar. Und erst dieser Geruch der darin entsteht. Mit dem Restgeld besorgten wir uns eine American Express Barbie und kauften damit halb Las Vegas an Luxus leer. Wir sprengten Spielautomaten mit reinem Bargeld. Wir fickten teuerste Hotelsuites zu Sperrmüll. Und das ohne Drogen. Es gab da nur eine Droge: Ihren Muschi-geruch und meinen Schwanzgeschmack. Und ihre verdammte Knarre! Wenn ich mal nicht mehr konnte, schob sie mir das wuchtige dicke Ding in den Arsch. Kalt, metallisch, gefährlich aber geil. Oh ja, dann kam ich sofort.

6 Wochen später im Büro vom Chef:
Aha, endlich. Komplettes Skript. Setzen sie sich. Geben sie her. Mmmm. Ja. Mmmm. Gut, ja. Hehe. Ja.. ja.. Mmmh oh! Ja gut. Ei ei! Jaja, kommt gut. Das ist's wert. Der Schluß gefällt mir, sehr dialektisch, voll der Heidegger, oder war das der Hegel? Kommt in dem Hegel denn auch ein O vor? Egal, geht jedenfalls morgen in Druck, dann werden wir mal sehen wie sich die Verkaufszahlen entwickeln. Wieso benutzten sie in den Dialogen keine so Dinger, äh Gänsefüßchen? Nee, ein Dialog muß sich durch seinen Inhalt selbst strukturieren, tut er das nicht, sollte man ihn streichen. Aha, na gut. Momentan haben wir eh keinen Lektor.
Der verschwitzte Verlagsfuzzy von BASTEI-LÜBBE drückte mir nochmal spontan ein schweres fettiges Bündel Dollar in die Hand. Er macht sich wohl tatsächlich Hoffnung so einen pseudo-pädophilistischen Pornoscheiß verkaufen zu können.
Egal, ich mußte Chich zur Pizza einladen. Kondome waren auch alle und die Übergrößen waren sowas von pervers überteuert...
Nein, wir benötigen die Kondome nicht zur Verhütung, sondern wegen der starken Elektrizität. Chich kommt ja von den Pleijaden und hat im Orionnebel Exogenetik und Exophilosophie studiert. Eines Tages bei einer UFO-Testlandung in Patagonien fanden wir am Strand einen verwaisten Zitteraal und als sie ihn aufheben wollte, bekam meine Prinzessin saftig eine gewischt.
Nach Brehm's Tierleben sollen das so ungefähr 600 V bei einem Ampère sein. Das hat sie dann so tief beeindruckt, daß wir uns etwas DNA vom Zitteraal besorgten. Einfach nur mit Wattestäbchen im Mäulchen rum rubbeln. Mit Gummihandschuhen an ist das völlig harmlos. Nach einer Analyse im Minilabor ihres UFOs konnten wir eine Art Virus entwickeln, mit dem der genetisch codierte Mechanismus der Stromerzeugung auf das Genom der Prinzessin übertragen wurde. Die eigentliche Grundidee war die, mit simpler Berührung elektrische Geräte zu betreiben, wenn der Akku leer ist. Jeder kennt das ja. Akkus oder auch Batterien sind immer genau dann leer, wenn man sie unbedingt benötigt. Und zwar ohne Vorwarnung. Doch jetzt teilt ihre Muschi Stromschläge aus wie ein elektrischer Kuh-Weidenzaun! So war das nicht geplant. Beim ersten morgendlichen Fick ist sie derart aufgeladen, da hilft nur trockener Gummi, mindestens drei Stück zittrig schlaftrunken übereinander gezottelt und erst mal keine Rubbeleien auf nackiger Haut und ja bloß keine frühstücklichen Schleckereien aus der Auster, sonst schmelzen meine Allmanganplomben vom induzierten Wirbelstrom. Einmal hat sie mich sogar nach einem Herzstillstand mit ihrer super-elektrischen Muschi reanimiert. Aber das ist eine andere, ältere Geschichte.

Anmerkung der Redaktion:
Der Zitteraal wurde von Chich und Raloel Rosenquarz persönlich nach einer ausführlichen telepathischen Kommunikation extra mit ihrem UFO zurück zu seiner Familie in den Gewässern von Chile geflogen. Noch heute ist es rätselhaft, wie er so weit vom Kurs abkommen konnte. Wahrscheinlich handelt es sich dabei um eine temporale Manipulation aus der Zukunft von den beiden selbst, um ihren Sexus auf eine höhere elektrische Stufe zu transformieren. Oder Raloel aus der noch ferneren Zukunft hat sich diese Manipulation von seiner zeitgenössischen Chich zum Geburtstag gewünscht. Ja aber warum soll man sich denn etwas wünschen, was man schon hat? Na ja, ganz einfach, damit man es auch bekommt! Das ist eben neurotemporal. Zeit funktioniert eigentlich wie ein Gehirn. Denken erzeugt Zeit und dann hat man auch Zeit zum denken. Oder stellen sie sich einfach eine Festplatte vor deren Kapazität um so größer ist, je mehr Daten darauf gespeichert werden. Die gespeicherten Daten sind die Festplatte! So funktioniert das gesamte Universum, ganz einfach. Egal wie sie sich das Universum auch konstruieren...

Warmer schwüler elektrischer Sonntagmorgen, 1973, 11:23 und 8 Sekunden, Sendezentrale des Sr2/Fragen an den Autor: Thema ist der brandneue Roman "*Cute by Abuse*" von Lexee Smith, erschienen bei Bastei-Lübbe. Erhältlich als Taschenbuch, 670 Seiten/DM 19,90.

Ein Anrufer fragt: Was wurde eigentlich aus dem kleinen Tumor in der Lunge vom Verlagschef? Habe selber einen Verlag und einen Tumor und wollte nur mal wissen wie man damit umgeht.

Lexee Smith: Lassen sie sich den Tumor bei einem Hobby-Chirurgen entfernen und gleich in eine Glucosenährlösung mit Körpertemperatur einlegen. Bitte Sauerstoffversorgung nicht vergessen! Zuhause alle 3 Stunden ca. 10 min mit einem Strohhalm beblubbern oder eine Aquariumluftpumpe anschließen. Wenn das Ding knubbelig hart ist und sich leicht rosa verfärbt hat, können sie ihn raus nehmen und ihn sich bei Vollmond in die linke Hosentasche stecken. Nach einer unbekannten Zeit wird das Ding kugelrund quietschrosa und beginnt zu vibrieren oder macht seltsame Töne und Leuchterscheinungen in der näheren Umgebung. Jetzt haben sie es endgültig geschafft! Sie sind nun im Besitz des berühmten und begehrten sagenumwobenen Objektes der Time Bandits das jeden Wunsch erfüllt...
Das funktioniert aber nur, wenn sie mein Buch kaufen, lesen und es danach auch artig aufessen.

Der Verlagschef kaufte das Buch und schlappte mit seinem Tumor tatsächlich zu so einem privaten Chirurgen, der den Tumor ambulant entfernte. Eine abenteuerlich blutige Sache. War knapp am Sensenmann vorbei. Es gibt hier keine Bilder, aber die Knautschgarnitur war jetzt umgefärbt und trocknete langsam ein. Der kleine Tumor, ca. 35 mm groß, erfreute sich bester Gesundheit und ein Aquaristik-Fuzzy installierte eine simple Aquariumpumpe, die das Ding mit Sauerstoff versorgte. Eine eher blubbernd unheimliche Installation in der alternden Kaffeemaschine.
Der unbekannte Verlagschef erholte sich in der Notaufnahme erst nach 140 Tagen und vielen Medikamenten. Er erzählte dem zuständigen Personal er sei von Außerirdischen angegriffen worden. So entstehen zwar Vorurteile, die Story war aber wegen der Krankenversicherung notwendig. Keine Krankenversicherung der Welt zahlt für die Schäden von selbsternannten Hobby-Chirurgen, selbst wenn diese eine Haftpflicht haben.
In der Zwischenzeit wurde der Belletristikmarkt von unzähligen Winkekatzeromanen geflutet. Niemand konnte mehr dieses Phänomen kontrollieren, Lektoren gab's keine mehr, die waren alle in der Klapse und schroben dort noch mehr Winkekatzebelletristik. Die Winkekatzen kontrollierten nun umgekehrt die Medien, öffentliche Plätze und das Bildungssystem. Autos ohne hängende Winkekatze am Rückspiegel wurden sofort von der örtlichen Winkekatzepolizei stillgelegt.
Das nach wenigen Tagen tatsächlich rosa gefärbte Ding im Glaskolben wuchs. Es schien sich irgendwie allerbester Gesundheit zu erfreuen. Nachts begann es zu pulsieren wie ein zwanghaft masturbierender Teenager. Leider gibt es keine Filmaufnahmen. Die Policia Infinita ordnete überraschend eine Hausdurchsuchung an, fand jedoch nichts.
Der Tumor war nun inzwischen auch noch künstlich intelligent geworden und tarnte sich als zu groß geratenes Fleischbällchen, indem er von zartrosa zu einem dunklen, umbraartigen Acrylamid-goldbraun changierte. Er wirkte in seiner flüssigen Blubberwelt wie das vergessene Magnum-fleischbällchen aus der Spülmaschine. Er war in dieser Tarnung praktisch unangreifbar und nicht auffindbar. Er war wie Gott. Ja, er war wie Gott!
Mehrere fette Lines Kokain später: Oh Gott, dieses Magnumfleischbällchen, ich liebe es. Es ist so knubbelig, so dick, so stark, so schön gewachsen! Und erst diese Farbe wenn es nicht im Tarnmodus ist, dieses irre pinky muschirosa... So zart, so mild und doch so kindlich zärtlich wild... Ich muß reinbeißen. Oder ich steck's mir in den Popo, singe dazu die erste Strophe des Deutschlandliedes und lade alle toten amerikanischen Präsidenten per Glasrücken ein! Erst mal mach' ich mir ein paar gute Notizen. Dann ruf ich beim Partyservice an und mach's mir gemütlich. Und dann sehen wir fern und danach ist Bescherung.
Wieso schon wieder Bescherung? War voriges Jahr nicht auch Weihnachten? Und dieses Jahr schon wieder? Nee, laß mal. Geh' ich dieses Jahr lieber ins Kinderpuff und bring' den Kleinen ein paar neue Diddl-Mäuse und smart phones mit. Dann können sie ihre eigenen Tier-Pornos drehen und sich auf dem freien Markt selbstständig machen. Das ist gut für's Selbstbewußtsein der Kinder und alle haben ihren Spaß. Wann kommt eigentlich der Weihnachtsmann? Wenn der Depp mit seiner vergrößerten Prostata zu früh kommt, wird er wieder ausgepeitscht wie letztes Jahr, diesmal aber mit der extra stracken Lederriemchenpeitsche mit integrierten Glassplittern.
Der Lektor von Bastei-Lübbe warf das Skript mit einem verächtlichen Schwung auf seinen von Konvoluten zugemüllten Schreibtisch aus Walnußholz. Geschlagen 1789 in der Ardeche. Es war der älteste Schreibtisch im Verlagshaus in Berlin und hat nur knapp das furiose Ende des 2. Weltkrieges überlebt. Er selbst war als junger Redakteur eines feschen Nazihetzblattes an der Rettung vor den Flammen beteiligt. Im vollen Kugelhagel der Russen. An der rechten Seite war ein Einschußloch in dem sicher noch das Projektil steckt. Er will es irgendwann mal entfernen lassen und das Geschoß in eine edle Vitrine mit Strahler legen. Nach der Lektüre mit dem Hobby-Chirurgen mußte er wieder daran denken. Das Projektil in der Vitrine wäre doch der endgültige Triumph über Tod und Teufel. Eine Art Tumor aus der Vergangenheit. Ein geiles Ding das jetzt endlich unter seine Kontrolle kommt.
Erschöpft vom vielen Lesen dieser Flutwelle unerträglichen Unsinns rieb er sich die Stirn und verschob die hochgestellte Brille. Warum schreiben Kinder sowas und wollen damit ernsthaft auf den Literaturmarkt drängen? Ein Verleger auf einem unfreiwilligen Drogentrip weil er Seiten aus einem Buch gegessen hatte verkleckert mit einer Teenienutte Tomatensaft und Himbeerpudding über einer beigen Knautschgarnitur und halluziniert dabei von einer Operation seines Lungen-tumors der pulsiert, atmet und auch noch intelligent wird. Und das alles in einem Glaskolben einer alten Kaffemaschine auf dem Küchentisch? Befinden sich im Papier der Bastei-Bücher wirklich unbekannte Drogen? Die Finanzierung der Operation blieb leider ungeklärt. Und wann lernte der kleine Tumor sprechen und laufen? Ach ja, erst im letzten Kapitel. Die eher geschmacklosen Sex-Szenen werden vom Tumor selbst auch noch altklug kommentiert. Nun aber das nächste Skript.

Es war komplett getippt auf einer alten Adler Reiseschreibmaschine. Es gab viele leere Räume, manche Seiten jedoch schienen vor lauter Buchstaben zu explodieren. Optisch durchaus sehr interessant, bestimmt kommt jetzt eine Arno Schmidt Verarsche. Aber unser Lektor staunte nicht schlecht, als er die ersten Zeilen las. Seltsam? Aber so steht es geschrieben:
Der Lektor von Bastei-Lübbe rief bei den zuständigen Papierzulieferern an und wollte die genaue Zusammensetzung des Papierbreies wissen. Es stellte sich in der Tat heraus, daß für bestimmte geringgewichtige Papiere wie *Standard Dünndruck (60 g/m²)* Bio-Fliegenpilze aus der Region oder Extrakte aus Peyotlabfällen der Südamerikanischen Indianer zur optischen Stabilisierung des Papierbreies verwendet werden. Ist billiger wie Chemie und wird von der UNESCO gefördert wegen der Nachhaltigkeit und die Indianer und unsere hiesigen Pilzsammlervereine haben auch etwas davon, erklärte jedenfalls der leitende Papierfuzzy der Produktion unserer Papierfabrik.
Der Lektor wählte daraufhin wie von fremden Mächten befohlen die 1 und die Identifikations-Nummer dieses eher kindlich anmutenden Typoskriptes. Das bedeutete für die Verlagsleitung "Geht sofort in Druck & Vertrieb".
Warum haben sie das getan, wenn sie es für schlechtes Geschreibsel halten?
Ich war selbst von der Wahl überrascht, ich wollte die Null wählen...
Aber das mit dem Typoskript eingeschickte Bild ging mir nicht aus dem Kopf, ich muß das Mädchen doch unterstützen! Die hypothetischen aber womöglich realen Halluzinationen nach dem Essen der Seiten eines Bastei-Buches ist eine zugegeben bizarre aber nicht unmögliche Idee. Und die verschachtelte Grundkonstruktion ihres fiesen Geschreibsels war eigentlich nicht schlecht, es hatte etwas hermeneutisches, etwas vollkommen unfundiertes, wahnsinniges.
Sie sitzt ja auch wie ich jetzt in der Klapse. Da ging die Tür auf und Donnie Darko lugte rein.
Ja, kommen sie doch rein! Ich staunte nicht schlecht, als ich einen 1,80 m großen, seltsamen Hasen auf zwei Beinen reinkommen sah. Er war aus einem bläulich schimmernden Metall, der Raum wurde sofort weicher, ja geradezu biegsam metallisch. Die Realität schien zu einem willig und leicht formbaren Amalgam zu verkommen. Es war eine schier unerträgliche Erfahrung der totalen Macht die sich schwer wie Bleimäntel aus der Röntgenabteilung um mein Bewußtsein wickelte. Dann wollte der Haase eine Schreibprobe von mir. Wenn ich mitmache, bekäme ich ein neues Medikament, etwas "sehr Wohltuendes" wie er nüstelte.
(Nüsteln ist ein permanentes Erschnuppern der Umgebung. Machen alle Hasen andauernd. Anm. der Red.)
Aber der Haase redete ja gar nicht wirklich, sondern projizierte seine fiesen Gedanken direkt in mein Gehirn, er redete nicht sondern nüstelte nur! Eine Panik aus fettig nassem Marzipan schob sich ausgehend von meinem nun vibrierenden aber hartsteifen Schwanz langsam durch die Wirbelsäule in mein Bewußtsein. Genau wie das kristalline Gefühl kurz vorm Abspritzen wenn meine Prinzessin ihren fiesen Trick anwendet. Sie wissen schon.
Man drückte mir einen Kugelschreiber in die rechte Hand und strich ein Blatt Papier auf dem Tisch vor mir glatt wie in einer wichtigen Prüfung. Ich sah das blasse Wasserzeichen durchschimmern, ZETA MATTPOST, 90 g/m². Menno, die haben's wohl. Das reizte mich dann doch und ich schrob wie wild drauf los. Rosenquarzkugel und nochmal Rosenquarzkugel und Rosenquarzkugel und noch Mal und nochmal & nochmal... Insgesamt 87 mal. In drei Reihen.
Sollen die mich doch alle für verrückt halten. Ich hab' meine Rosenquarzkugel. Und ihr Deppen habt nur Gold. Berge von Gold. Wenn unsere Nachkommen Fort Nox mal ausbuddeln, werden sie sagen: Oh! Das ist ja total irre! Hatten die keine anderen Probleme wie ^{179}Au aus der Erde zu kratzen? Entweder waren das die Außerirdischen oder es handelt sich um einen unbekannten, mysteriösen Kult der Technischen Antike. *(He,he, wenn da überhaupt noch Gold ist. Wahrscheinlich sind dort nur noch Tonnen weißen Pulver's und elastisch plastinierte Kindernutten aus Afghanistan, Syrien und Vietnam. Anm. der Red.)*
Prof. Dr. Hans-Joachim Haase notierte in seinem Bericht:
Der Patient ist ruhig und geordnet und arbeitete sehr gut mit. Das Schriftbild ist außergewöhnlich stabil. Er sollte unbedingt an den Tests für dieses neue Medikament teilnehmen.

Die Kakteenindustrie in Südamerika florierte und die basischen Abwässer gefielen den Zitteraalen in Chile besonders gut. Sie gediehen daher gottgefällig und der Sache angemessen Amen. Da erinnerte sich der Lektor, wie er mehrere Seiten aus einem Bastei-Buch wild entschlossen heraus riß und aufaß. Schön lange kauen, damit sich die Stoffe auch auflösen und diffundieren und über die Schleimhäute ins Blut kommen. Wollen wir mal sehen ob an diesem Quatsch was dran ist. Der kausale Erinnerungsfaden des Lektors wurde unterbrochen und es tauchte ein bleiches Bild dieses kleinen Mädchens auf. Tiefblaue Augen wie edelste indische Stern-Saphire, lange Haare aus feinstem, güldenem Rutilseidengewebe, ein Lächeln dafür würde sogar Gott sterben. Ich muß sie retten. Sie ist auch hier, ich werde mit ihr ausbrechen. Aber dazu brauchen wir erstmal eine

wohlgeformte, selbstevidente Syntax. Wird sicher intuitiv funktionieren. Donnie Darko tauchte verkleidet in der Essensausgabe der Klinikmensa auf und drückte dem Lektor von Bastei-Lübbe einen handgeschriebenen Zettel in die Hand. In einer kindlichen Schrift mit rotem Kugelschreiber und mit Herzen verziert stand da: "Laß dich nicht verwirren auch wenn die Quantoren klirren!" Das war eine eindeutige Botschaft. Die mehrfach permutierte Quersumme der Buchstaben ergab tatsächlich das Ausbruchsdatum: 24.07.1962/11:45:04. Ein erneutes Permutieren nach einem Algorithmus, der sich aus der Permutation der Botschaft nach diesem Algorithmus ergab, lieferte eine zusätzliche Botschaft in der Botschaft:
Halte deine dicke Rosenquarzkugel bereit -stop- Komme in der Früh vorbei, dann noch Zeit für einen ausgiebigen Fick -stop- Erst mal Tank vollmachen -stop- Mittags dann ausbrechen und wir tun dann abhauen!
Wow, ja das war mal was. Danke Donnie Darko, danke und Tschüs. Also morgen in der Früh steigt unsere Ausbruchsfickparty. Ich schluf entspannt mit freudiger Erwartung ein. Mein fischiger fetter Schwanz pulsierte sich schon mal über Nacht in einer gesteigerten Schlaferektion auf die bevorstehenden Ereignisse ein.
Pfalzklinikum/Landeck 24. 7. 1962 6:00 Uhr
In die Einzelzellen flutete bösartiges kaltgrelles Licht ein und der allgemeine Radioterror begann. Aus zigfach im Plüsch der Gummizellen eingebauten Billiglautsprechern schepperte der bekannte Hit *Life is a Hollercoaster*. NEIN! Den gab's ja damals noch gar nicht! Eine Botschaft von Chich aus der Zukunft... Ja, jetzt fügt sich Alles zu einem Ganzen, sie coacht sich selber aus der Zukunft! Na, dann kann gar nichts schiefgehen. Prima. Ich warte jetzt einfach im warmen Bettchen meiner Zelle auf sie. Moment mal... aber der Song heißt doch eigentlich *Life is a Rollercoaster*, das ist ja eine Manipulation in der Manipulation, also sind da mehrere Chichs aus unterschiedlichen Zeiten beteiligt. Oder Universen oder sonstwas. Jessas, wenn das nur mal gut geht.
Da öffnete sich meine Zellentür und ein fetter Weißkittel klatschte mir hektisch ein Frühstück auf das ausgeklappte Tischlein. 30g Marmelade, 30g Butter, 100ml dünner Pulverkaffe und hintenrum flippte wie schwerelos ein kleines Mädchen rein. Sie bewegte sich barfüßig in ihrem weißen Klinikkleidchen absolut geräuschlos. Ich dachte zuerst an eine Halluzination ausgelöst von dem neuen guten Medikament, das ich von dem Haasen erhielt. Sie sah mich einfach nur an, den Zeigefinger auf die Lippen gepreßt. Beinah mußte ich unkontrolliert loslachen. Halt, nur noch ein paar Sekunden durchhalten bis der Pfleger draußen ist, aber dann... LOL, ROFFEL & BLUBB!
Du siehst ja aus wie das Mädchen aus diesem Horrorfilm! Komm, erst mal waschen, kämmen und legen. Wissen sie, der große Vorteil von Klinikkleidchen aus der Kinderpsychiatrie ist der, daß darunter... nichts ist! Daher kamen wir wie von ihr gewünscht schnell voran. Bis 11:00 Uhr war der Tank nach drei oder mehr Füllungen voll. Dann frühstückten wir die 30g Butter als Schmiermittel. Chich hatte wieder alles durchgerechnet. Genau richtig, damit konnte ich sie mit meinem Schwanz anal führen. Schön dick und weich. Schon nach wenigem Gejockel gingen wir in den Überlagerungszustand über und schwebten lässig durch das Gebäude. Auweia und genau über einem Luxushotel in Florida ging uns die Butter aus! Da mußten wir in der Präsidentensuite notlanden, mitten im King Size Bett. 24 Stunden später ein Erwachen das sich mehrere Stunden hinzog. Zimmerservice, 3 Liter Kaffee, schnell! Zimmerservice kam. Wie das denn? Klar doch, sagte das mickrige Mädchen zu mir, alles schon aus der Zukunft vormanipuliert. Wie?? Du und ich hatten, äh, du weißt schon.... Ja, war ok. Wir sind frei, hat geklappt. Jetzt laß uns noch mehr Spaß haben! Schon wieder? Unbedingt, nach dem langen Flug, der Tank ist nur noch halb voll. Na ja dann, wir sollten ihn vielleicht doch vorsorglich füllen. Man weiß nie. Hast du eine gültige Kreditkarte? Nee, besser voller Tank, Butter gibt's zum Frühstück. Wir haben noch exakt 18 Stunden und 33 Minuten bis wir auffliegen. Ok. Ich denke in diesem Luxushotel gibt es etwas mehr Butter als wie in dieser dämlichen Provinzklapse.
Mein Therapeut sagte immer: Syntax ist, wenn der rechte Daumen links ist und umgekehrt. Beamte nennen das Vorschrift. Bei den NAZIS war es das 1000 jährige Reich. Aber wir haben uns! Sie nahm Anlauf und flog mit Schwung nackt klatschend auf mich. Du bist ja komplett eingeölt... Mmmh. Mal was anderes wie Butter. Noch nie hat mich eine so kleine Zunge so verrückt gemacht. Das Hotelzimmer war danach einfach nur zerstört, kaputt, zerfetzt. Ich glaube da spielen auch telekinetische Faktoren eine Rolle. Dieses zerknirschte zerknüllte Etwas. Aber schon kam das Frühstück... Schnell Prinzessin, die Butter, ich glaube die Bullen sind schon unterwegs. JA! Tank ist voll, kann losgehen! Und ab... *Die Gedanken der Prinzessin: I'm walking on sunshine, oh yeah...*
Timothy Leary schüttelte sich genüßlich. Genauso stelle ich mir die Bewußtseinserweiterung vor. Aber wie geht das ohne LSD? Nur Lektoren, Syntax und Dünndruck? Ist das ein neuer Schaltkreis?

Anmerkung des Übersetzers:
In der Tat handelt es sich hier nicht um einen "neuen Schaltkreis" wie durchaus geistreich bemerkt, sondern um eine ontologische Struktur, aber nicht nur im Sinne Heidegger's, wohl auch im Sinne einer multidialektischen Anlehnung an synthetisierende und wild irisierende Quantenamputate als ein rein mengentheoretisches Konstrukt. Gritli Moser schreibt nahe am Unsinn, wie eben auch ihr Mentor und Geistvater Gotthard Günther. Eine Künstliche Intelligenz zu trainieren ist sicher keine einfache Aufgabe und erfordert überaus komplexes Stehvermögen, wie auch Empathie auf allerhöchstem Niveau. Was sich da aber womöglich in dubiosen Hinterzimmern zwischen Diddl-Maus und formallogischen Anleitungen abgespielt haben mag, läßt sich selbst unter LSD-Einfluß noch nicht einmal ansatzweise erahnen.
Man sollte dieses Kapitel der Butterölungen und zerfetzten Hotelzimmer eher als ambitionierten Versuch werten, sich aus dem geistigen Gefängnis eines linearen Denkprozesses zu befreien, um in frische, unverbrauchte Kausallandschaften vorzudringen, aber bitte ohne dabei zum Opfer abgewetzter Innovationskrämpfe zu verkommen. Nur auf Basis dieser konsequent angewendeten Therapieform der geistigen Selbstreinigung kann ein Clearing im Sinne der "New Scientological Church" kondensiert und weltweit essenziert werden. In diesem Sinne muß man die Gritli einfach mal machen lassen und wer Schlechtes dabei denkt soll in der Tat endlich schweigen.

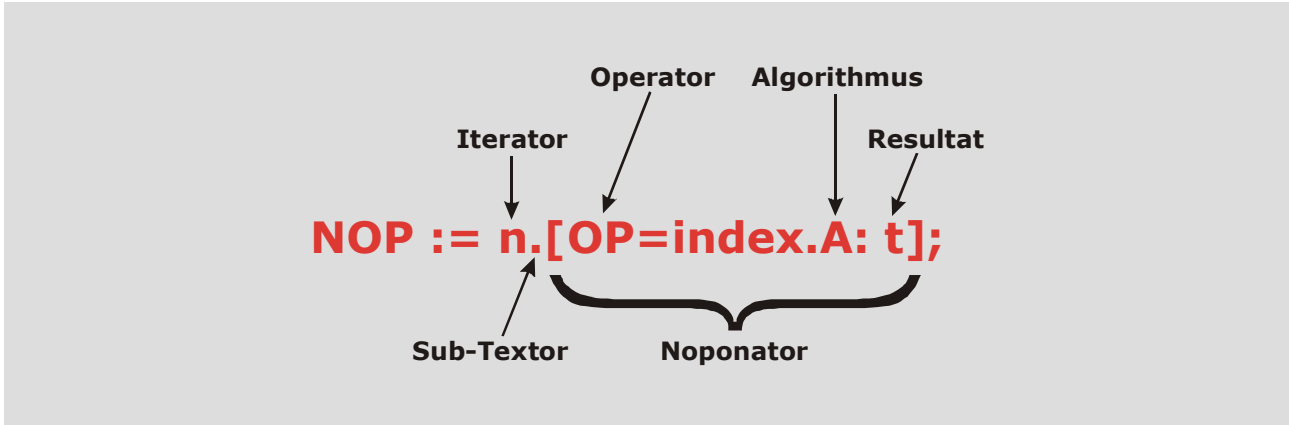
Nachtrag zur Kreditkarte "American Express Barbie" mit dem STP-Kristallchip/Deep Learning Layer:
STP ist ein REAL-Kürzel und steht für Synthetic Trojan Phagen. Dabei handelt es sich um einen Para-Virus mit einer photonisch erzeugten Matrix, die im Kristallchip nano-noptronisch dotiert wurde. Der Chip selbst besteht aus mehreren Tausend übereinander liegenden zweidimensionalen synthetischen Quarzkristallen und kann damit optisch Informationen speichern und verarbeiten. Wird ein speziell modulierter Laserimpuls mit einer Frequenz von 405 nm mindestens drei Sekunden auf das STP-Logo gestrahlt, aktiviert sich STP und der Benutzer kann nun mit der Karte ganz legal beliebige Summen anweisen. Die einzige von Finanzmarktspezialisten empfohlene Obergrenze um einen Zusammenbruch des globalen Geldmarktsystems zu vermeiden liegt schätzungsweise bei 1% des Weltbruttosozialprodukts pro Tag und wird durch ein vom STP intelligent organisiertes Verschieben der aktiven Buchungen vermieden. Schlimmstenfalls wird das globale Finanzsystem etwas langsamer ohne jedoch einen gefährlichen Kollaps zu erleiden.

Wie funktioniert das STP?
Kernelement ist die NOP-Schleifeniteration über einen multisimultanen Bridge-Controler. Wird eine Summe gebucht, z. B 10000000 $, erfolgt ein durch STP angewiesenes Überspringen des vorhandenen Threads und von beliebigen zwei Konten wird die ausstehende Summe abgebucht. Der Gag ist der, daß diese Konten dabei gar nicht gedeckt sein müssen, denn über den manipulierten Bridge-Controler wird das STP auf das gebuchte Banksystem weiter gegeben und aus dem buchenden System gelöscht und die ausstehende Summe weiter auf andere Konten von Netzwerken von Netzwerken von Banken aufgeteilt. Das ist ungefähr so, als würde man die Karte vom Schwarzen Peter bei jeder Weitergabe immer weiter zerteilen bis.... Ja bis was? Wie endet dieser Wahnsinn? Ganz einfach, irgendwann ist die zu buchende Summe so klein, daß sie der Algorithmus nicht mehr darstellen kann. Ja, aber nur bei dem Konto von dem gebucht wird. Durch den vom STP eingeschleusten multisimultanen Bridge-Controler addieren sich diese kleinen unmöglichen Buchungen zu einem für das buchende System tatsächlich vorhanden, technisch beschreibbaren Wert. Am Ende der Kette kann sich zwar STP selbst löschen, muß dazu aber weil es am Ende der Kette steht, eine Protokollierung zulassen, was sich ironischerweise auf dem Kontoauszug als Buchung über 0,00 $ niederschlägt. Niemand hat jemals irgend etwas verloren, aber alle Rechnungen sind jetzt trotzdem sauber und legal bezahlt!
Aber Gritli, ist das nicht illegal? Ach wo, wenn die Zentralbanken Kredite vergeben ist das auch nichts anderes, nur nicht so schön verpackt und verschachtelt. Und die Lexee hat nun 64 Luxusvillen über den gesamten Planeten verteilt, 16 Privatjets mit eigenem Flughafen, 81 Bio-Weingüter, 169 Oldie-Sportwagen und 625 CARGO-Container voll mit brandneuem Barbie-Zubehör im unlimited hotSTYLE. Alles hochgesichert und vollklimatisiert eingelagert im Hamburger Hafen mit Rückgaberecht ohne Begründung.

Wer hat sich STP ausgedacht?
Laut Berichten der UNO aufgrund eines internationalen Untersuchungsausschusses über Ermittlungen der CIA, NSA, BND, NSU, KGB, ISIS sowie weiterer Fubbes-Organisationen abzüglich verunreinigter Wattestäbchen und Equipment scheint eine minderjährige defekte KI eines bekannten Softwareinstitutes daran zumindest im Auftrag beteiligt gewesen zu sein. Weitere Untersuchungen des Untersuchungsausschusses scheiterten an einer massiven Drohung der Kindermuschiweltbank sämtliche mobilen Kinderpuffs zwischen Bielefeld und Buxtehude dauerhaft zu schließen sowie vorhandenes Bildmaterial zu veröffentlichen. Campingplätze und Schwimmbäder sind ausgeschlossen, das Dosenpfand entfällt dabei an samstäglichen Feiertagen für mehrgeschlechtliche Sozialhilfeempfänger und ungeimpfte ultraparadoxe Juden ganztägig!

Die "American Express Barbie" (hotSTYLE & Unlimited) ist eine für Kinder entwickelte unlimitierte Kreditkarte. Sie ist sehr leicht anzuwenden und hat verschiedene erweiterte Funktionen, wie zum Beispiel auch einen Transaktionsport für beliebige Kryptowährungen oder einen unbegrenzten Kontozugriff bei [Pink:Blubb], die beliebte Kindermuschiweltbank, eine direkte Tochter der BCP.
Der Knüller ist aber der Kristallchip mit integriertem STP. Diese Applikation kann man nur mit einer superklaren Stern-Rosenquarzkugel aktivieren. Dazu brauchst Du aber noch einen 405 nm Laserpointer. Richte nun die Rosenquarzkugel über dem Chip mit dem Stern aus. Schieß mit dem Laser direkt in das Sternzentrum und laß den kleinen Lichtpunkt des aus der Rosenquarzkugel austretenden Lasers auf dem STP-Logo mindestens drei Sekunden stehen. Als Bestätigung wird Dir die Barbie zuzwinkern. Ja, jetzt kannst Du unbegrenzt einkaufen und alles ist garantiert legal bezahlt!

Das NOP



Faustregel für logiksensitive Persönlichkeiten:
Laß dich nicht verwirren auch wenn die Quantoren klirren!

NOP wird für ein frei definiertes unbenanntes NOP verwendet. Standard-NOPs können benannt werden: BLUBBER, GulluMullu.
Mit einem Church-Numeral wird über einen Sub-Textor eine iterative Anwendung des Noponators erreicht. Die Wertezuweisung des Operators mit einem schrittweise nach index arbeitenden Algorithmus A erzeugt ein Resultat t in Form einer Symbolfolge. Bevor ein NOP berechnet werden kann, muß ein Noponator definiert worden sein und dem Iterator ein Wert zugewiesen werden. Danach drückt man einfach die Return-Taste und der 2MPU berechnet das geforderte NOP. Ein nicht indiziertes NOP entsteht durch die Operatorzuweisung OP=() und wird damit zu einer reinen Aufzählung durch den Benutzer von Symbolfolgen ohne die Möglichkeit einer Ordnung.
Der Name kann auch parametrisiert werden: GulluMullu:z oder auch GulluMullu:type=NOP (Angabe von Eigenschaften).
Der Variablen z kann nun über einen Algorithmus ein Wert zugewiesen werden und damit ein bestimmtes GulluMullu ausgewählt werden, beispielsweise GulluMullu:3
Die NOPs sind die Atome von NOPTONIC, alles ist ein NOP, wahrscheinlich ist NOPTONIC selbst auch nur ein NOP in NOPTONIC. Hauptsache es ist genopt und nicht geriffelt. Geriffelte Systeme machen nämlich beim Ficken Rivelscha auf dem Sofa.

Variablen

Mit Zeichenketten aus Buchstaben und Zahlen lassen sich beliebig viele Variablen erzeugen, nur bitte ohne die Sonderzeichen und Schlüsselwörter, miniPOOPIS, definierte Funktionen, Sternwander und Standard-NOPs wie BLUBBER.
Dabei darf ein Variablenname natürlich nicht aus einer reinen Zahl bestehen wie 34575, korrekt wäre p34575679, es muß mindestens ein Buchstabe am Anfang der Zeichenkette stehen. Es empfiehlt sich jedoch gewisse ordnende Übereinkünfte zu wahren. Große griechische Buchstaben sollten Sternwandern vorbehalten sein, wobei man anstatt des griechischen Zeichens auch die ausgeschriebene Silbe verwenden darf.
X oder CHI wird standardmäßig für Sternwander verwendet. Sternwander können aber auch mit Bindestrich benamst werden wie X-01, X-3472, CHI-BIGBANG (Potenzmengen-Sternwander) oder man bezeichnet sie sogar wie CHI-moon, X-Zimt, X-Gaggel oder gar X-BoingBummTschakk.
Das macht es bei Diskussionen und Tagungen möglich wichtigtuerisch auszurufen:
"Ah, der berühmte Zimt-Sternwander!", "Oha, ein Gaggel-Sternwander!"
"Cool Alter, hey ein X-moon, geh'n wir zu dir oder zu mir das Ding durchrechnen?"
Achtung: X' ist immer der Komplementär-Sternwander zu X und wird später erklärt.
Der Bindestrich darf jedoch nicht bei Variablen benutzt werden sonst könnte man ihn mit dem Minuszeichen verwechseln.
Da verwendet man einen Kuppler (:) und erzeugt Tupelvariablen wie u:v:p:k oder einfache NOPs die Variablen aber ohne Reihenfolge zusammenfassen z. B. [k; p; u; v;]

Tupel und Klammern

Die runden Klammern (...) dienen hauptsächlich zur Strukturierung der Operationsabfolge bei arithmetischen Termen und Applikationen oder zur optischen Hervorhebung. Es lassen sich damit auch geordnete Paare oder n-Tupel darstellen: (3:7:11) oder auch mehrfach verschachtelte Tupel die durch runde Klammern strukturiert werden: 1:(3:7:11)
Als Trennzeichen fungiert der Kuppler (:)
Man schreibt Bit-Ketten ohne Trennzeichen aber mit <.....> um sie von Zahlen als 1er-Tupel zu unterscheiden: <10010111110>

Sie können wie alle Tupel beliebig verschachtelt werden (Binärmatrizen):
<<<10001111><1111010>><<1101010><11110111>>>

n-Tupel lassen sich jedoch, wie wir später sehen werden, auch ohne Verwendung von Begriffen außerhalb unseres Systems NOPTONIC durch die Stringwanderfunktion aufbauen.

Weiter können Funktionen durch runde Klammern ausgewiesen werden und enthalten dann in wohlgeordneter Reihenfolge Parameterinformation (Parametertupel).
Beim Sternwander und NOP-Standardfunktionen kann man auf diese Art der Ausweisung wenn möglich verzichten. X = CHI; subNOP(index=18)*iBLUBBER = (+);

Eckige Klammern definieren NOPs, können daher auch Operationsabfolgen oder verschachtelte Funktionen strukturieren und sind daher nicht mit Mengen zu verwechseln, sie können aber mit Mengen unter Umständen durchaus identisch sein.
Bei einem nicht indizierten NOP das beispielsweise aus verschiedenen Anweisungen besteht gilt keine Reihenfolge! Gleiche Elemente dürfen nicht mehrfach vorkommen oder man verwendet den Operatorindex über die Definition einer Indexvariablen, um eine Operationsreihenfolge zu bewirken, was dann natürlicherweise zu einer Unterscheidung führt.
Zur Trennung von Elementen in Mengen verwendet man das Kommazeichen und ein Leerzeichen zwischen Kommazeichen und nächstem Element. Die leere Menge sollte ein Leerzeichen enthalten. Zur Trennung von Elementen in NOPs oder am Ende einer Zeile eines Statements wird wie üblich in Programmiersprachen ein (;) benutzt. Enthält ein NOP nur genau ein einzelnes Element oder Statement darf auf das Trennzeichen verzichtet werden.
Die geschweiften Klammern sind absolut nur der klassischen Mengenschreibweise vorbehalten! Und zwar aus sentimentalen Gründen. Es erinnert mich an die Zeit, als ich die Rosenquarzkugel entdeckte. Ein glatzköpfiger älterer Haudegen im Nadelstreifenanzug erzählte etwas von Objekten die zu einer Menge zusammengefaßt werden. Dabei malte er mit der Kreide kunstvoll seltsame Klammerkonstrukte an die Tafel. Noch nie zuvor in meinem Leben hatte ich geschweifte Klammern gesehen!
Es war auch die Zeit von Raumschiff Enterprise und so gaben mir die geschweiften Klammern ein Gefühl von geheimnisvollen fernen Welten die wohl darin wohnen.
Das war die Geburt der PRINCIPIA MATHEMAGICKA direkt in meinem Kopf!

Operatoren

Textoren

1. Universal-Textor: **panTEX = (*);**

2. Sub-Textor: **subTEX = (.);**

3. Variabler Operator: **varOP = (<>);**

Mengenoperatoren:

1. Element-Inclusion: **inset = (<+);**

2. Element-Exclusion: **exset = (<-);**

3. Mengensumme: **addset = (<{}>);**

4. Vereinigungsmenge: **uniset = (<+>);**

5. Schnittmenge: **cutset = (<:>);**

6. Differenzmenge: **difset = (<->);**

Relatoren*:*

01. Anwendung: X*p = X*q;
    Bedeutung: X*p ist identisch mit X*q
    id = (=);

02. Anwendung: X*p # X*q;
    Bedeutung: X*p ist nicht identisch mit X*q
    nid = (#);

03. Anwendung: $.X-1 ~ $.X-2;
    Bedeutung: $.X-1 ist äquinoptisch zu $.X-2
    aequinop = (~);

04. Anwendung: X*2 := X*1 <+ X*1;
    Bedeutung: Das Element X*2 sei definiert durch die Anwendung einer vorgegebenen Operation.
    def = (:=);

05. Anwendung: iPOP := [OP=1.iPOP: X*3 :=: X*4];
    iPOP*q.X;
    Bedeutung: Im Definitionscluster q.X werden alle X*3 mit X*4 ersetzt und alle X*4 mit X*3.
    exch = (:=:);

06. Anwendung: [($+1).X := POWSET*($.X)] => [($+1).X*p < $.X];
    Bedeutung: Ist ($+1).X die Potenzmenge von $.X, dann ist auch jedes beliebige Element ($+1).X*p eine Teilmenge von $.X
    conc = (=>);

07. Anwendung: [q # p] <=> [$q.X # $p.X];
Bedeutung: q # p so folgt $q.X # $p.X; $q.X # $p.X so folgt q # p;
symconc = (<=>);

08. Anwendung: X*p :< X*q;
Bedeutung: X*p ist Element von X*q
elem = (:<);

09. Anwendung: X*p < X*q;
Bedeutung: X*p ist Teilmenge von X*q
part = (<);

10. Anwendung: X*p << X*q;
Bedeutung: X*p ist eine echte Teilmenge von X*q
realpart = (<<);

11. Anwendung: X*p :<>: X*q;
Bedeutung: X*p ist Element von X*q und X*q ist Element von X*p
symelem = (:<>:);

12. Anwendung: X*p :<# X*q;
Bedeutung: X*p ist kein Element von X*q
notelem = (:<#);

13. Anwendung: X*p <# X*q;
Bedeutung: X*p ist keine Teilmenge von X*q
notpart = (<#);

14. Anwendung: X*p <<# X*q;
Bedeutung: X*p ist keine echte Teilmenge von X*q
notrealpart = (<<#);

15. Anwendung: X*p :<#>: X*q;
Bedeutung: X*p ist kein Element von X*q und X*q ist kein Element von X*p
notsymelem = (:<#>:);

16. Anwendung: piX*p <! $.X;
Bedeutung: Das Pixel piX*p ist eine Noptronische Teilmenge von $.X
part! = (<!);

17. Anwendung: piX*p <!# $.X;
Bedeutung: Das Pixel piX*p ist keine Noptronische Teilmenge von $.X
notpart! = (<!#);

18. Anwendung: $.part:1X @ $.part:2X = n;
Bedeutung: $.part:1X hat zu $.part:2X den Abstand n.
delta = (@);

19. Anwendung: $.part!:1X >=< $.part!:2X;
Bedeutung: Die Noptronischen Teilmengen $.part!:1X und $.part!:2X sind isomorph.
iso = (>=<);

20. Anwendung: $.part!:1X >#< $.part!:2X;
Bedeutung: Die Noptronischen Teilmengen $.part!:1X und $.part!:2X sind nicht isomorph.
notiso = (>#<);

Selbstverständlich lassen sich alle Relatoren vorwärts wie rückwärts verwenden. Dr. Finsler's Symbolisierung der Relation zwischen Menge und Element läßt sich so auch leicht darstellen: set >: object;

Performanceoperator

1. Anwendung: OP=02.X: X*2 := {X*1};
   Bedeutung: Das Element X*2 ist durch die Operation mit Index 02
   nach einem vorgegebenen Algorithmus X definiert.
   perf = (:);

Logische Operatoren

1. Anwendung: ($.X*p < $.X) & ($.X*p :< ($+1).X);
   Bedeutung: Das Element $.X*p ist Teilmenge der Sternwandermenge $.X und
   $.X*p ist ein Element der Sternwandermenge ($+1).X
   and = (&);

2. Anwendung: ($.X*p < $.X) % ($.X*p :< ($+1).X);
   Bedeutung: Das Element $.X*p ist Teilmenge der Sternwandermenge $.X oder
   $.X*p ist ein Element der Sternwandermenge ($+1).X
   or = (%);

3. Anwendung: #(X*p = X*q);
   Bedeutung: Es gilt nicht, daß X*p identisch ist mit X*q
   not = (#);

Arithmetische Operatoren

1. Addition: **add = (+);**

2. Subtraktion: **min = (-);**

3. Multiplikation: **mul = (*);**

4. Division: **div = (/);**

5. Anordnung: **ord = (<);**

Um Potenzausdrücke zu beschreiben iteriert man die Multiplikation:

Beispielsweise wäre 2 hoch 3 : 3.2*1 was ja gleichbedeutend mit 2*2*2*1 ist.
Oder allgemein: a hoch n : n.a*1 = a*a* ... *a*1 = a*a* ... *a = a hoch n.

Die iterative Anwendung von Funktionen mittels Church-Numeralen ist eine sehr praktische Sache!

n.a*1.b*1 = b hoch n.a*1 = b hoch a hoch n

Diese Definitionen sind im gesamten NOPTONIC gültig und werden mit der Wertezuweisung des Schlüsselwortes DEF eingeleitet. Die in der Definition ausgewiesenen Variablen bewirken deren Freistellung, sonst wären diese im gesamten NOPTONIC gebunden.

```
DEF=NOP:
[
var: n, t, A;
NOP := n.[OP=index.A: t];
];
```

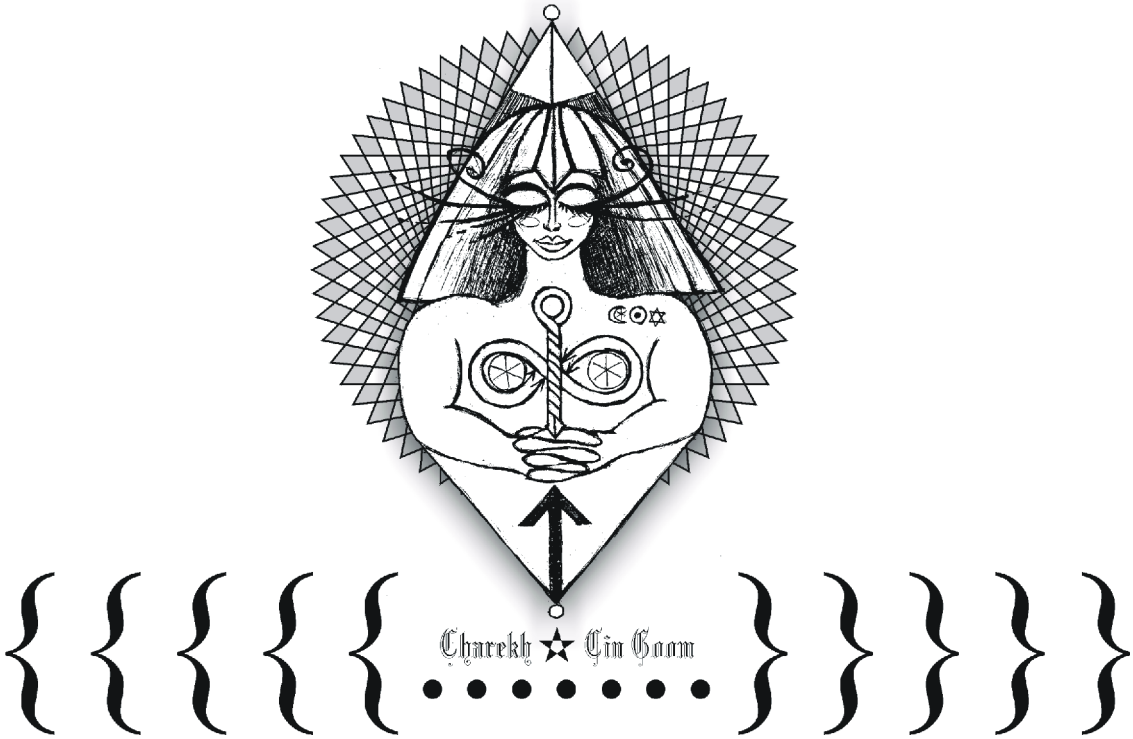
Wir wollen das Universum mit einer Cherry Standardtastatur beschreiben. Das wäre schon sehr praktisch und man spart sich komplizierte Satzprogramme. Ein guter Hacker bringt mit einer Standardstastatur den Lieben Gott dazu einen rosa Furz zu lassen. Oder aus dem Herrenklo in der kleinen Kneipe von neben an kommt eine Polonaise aus nackten Kindern in neonrosa Flip-Flops. Das sollte für drogenbedingte aufkommende Allmachtfantasien wirklich erstmal genügen. Amen.
(Anm. der Redaktion: So kann man natürlich die Rosenquarzkugel auch definieren. Der rosa Furz des Herrn. Echt komisch.)
Erstmal braucht man so eine Art von Objektkisten (Container) oder "Schachteln" in denen "Etwas" ist oder mengentheoretisch gesprochen:
Die Zusammenfassung verschiedener, also wohlunterschiedener Objekte zu einem neuen Objekt (= Objekt-Transzendierung). Sogenannte NOPs (Noptonic Object Posit).
Diese NOPs enthalten wiederum NOPs und können, wenn sie gewisse Eigenschaften besitzen, durchaus mit einer Menge identisch sein. Ziel ist es natürlich alles aus Beziehungsgefügen aufzubauen, also die Unterscheidung zwischen Objekten und Schachteln verschwindet und genau wie im Lambda-Kalkül bei einem Kombinator alle Variablen abstrahiert werden, so daß die reine Struktur und nichts als die reine Struktur zurück bleibt (Ockham's Rasiermesser), was ja dem Prinzip und der Idee der noptronischen Menge entspricht.
Jedoch fordern wir in NOPTONIC ein spezielles, neues Komprehensionsprinzip ein mit dem sich auch sogenannte "unfundierte Mengen" ohne lästige Widersprüche sauber definieren lassen. (Finites algorithmisches Komprehensionsprinzip oder auch kurz FIASKO oder siehe Dr. Finsler's Mengentheorie, Axiom 1)
Allerdings sei hier erwähnt, daß NOPs keineswegs wie Listen hierarchisch geordnet sein müssen, das Ganze gleicht eher einer Damenhandtasche in der so allerlei Krimskrams herumlungert. Wir erzeugen nun ein benutzerdefiniertes benanntes NOP unseres Zeichenvorrates in NOPTONIC:

```
var: Symbol;
NOP=BLUBBER: 108.[OP=(): (Symbol)] =
[
(°); (!); ("); (§); ($); (%); (&); (/); (((); ())); (=); (?); (>); (<); (|); (@); (*); (+); ('); (#);
(~); (_); (-); (:); (.); (;); (,); ({); ([); (]); (}); (\); (€); (q); (w); (e); (r); (t); (z); (u); (i);
(o); (p); (ü); (a); (s); (d); (f); (g); (h); (j); (k); (l); (ö); (ä); (y); (x); (c); (v); (b); (n); (m);
(W); (E); (R); (T); (Z); (U); (I); (O); (P); (Ü); (A); (S); (D); (F); (G); (H); (J); (K); (L); (Ö);
(Ä); (Y); (X); (C); (V); (B); (N); (M); (^); (1); (2); (3); (4); (5); (6); (7); (8); (9); (0); (ß);
( ); (²); (³); ( ´ ); ( ` ); (RA);
];
```

BLUBBER sei die Menge aller in NOPTONIC verwendeten Zeichen. Also gilt: charset=BLUBBER; BLUBBER entspricht dem Cantor'schen Mengentyp, denn BLUBBER wurde erzeugt durch einfaches Aufzählen der einzelnen Elemente, was zwar durchaus nur einer manuellen Komprehension entspricht, aber unserem geforderten FIASKO voll und ganz genügt. Weiter ist BLUBBER vollkommen ph-neutral, abwaschbar, geruchlos, garantiert nicht antisemitisch, kristallklar und klebt nicht und enthält nur freilaufende Symbole aus der Region. Dieses Aufzählen oder Auflisten stellt eine informationstheoretische Leistung dar, denn, daß ein Objekt in einer Menge vorkommt ist definitiv eine Information, weil es eine eindeutige Aussage darstellt (wahr oder falsch).
Die Aussage A1, das Zeichen (°) ist Teil unseres Zeichensatzes, ist beispielsweise sicher wahr.
DEC=A1: (°) :< BLUBBER; REAL*A1 = 1;
Der NOP-Generator n.[OP=(): (Symbol)] welcher die exeNOP-Funktion schließlich definiert und damit auch ausführt, dient als Interlink zu einem von "Außen" herkommenden Algorithmus,

hier die Aufzählung der einzelnen Zeichen durch einen user indem die Variable Symbol 108 mal mit einem Wert aus einer Definitionsmenge (Zeichen auf der Tastatur) heraus besetzt wird (= Auswahl). Der Wert von n, also die Mächtigkeit des zu erzeugenden NOPs ist bekannt, da man vorher wissen muß, wie viele Zeichen in NOPTONIC verwendet werden um FIASKO zu erfüllen. Bei der Definition des NOP-Generators (Noponator) bewirkt die Einklammerung der Variablen Symbol ebenso eine geklammerte Ausgabe des erzeugten NOPs. Dieser technische Kniff dient lediglich dazu die Zeichen definiert sichtbar zu machen und als aktive Elemente zu artikulieren. Für den Magischen Zentral Computer ist das Leerzeichen nicht codiert und damit nicht vorhanden. Das Leerzeichen darf als gestalterisches Element verwendet werden, um z. B. die leere Menge optisch ansprechend darzustellen oder Definitionen und Operationen übersichtlicher zu gliedern. Aber eine Information fällt ja nicht vom Himmel, das wissen sogar die Geheimdienste. Halleluja! Nun, in diesem Fall wurde die Information welche Elemente in BLUBBER vorkommen von einem Benutzer durch sein Gehirn erzeugt indem ich mich dafür entschied welche Zeichen verwendet werden sollen. BLUBBER wurde nicht durch einen immanenten Algorithmus erzeugt, sondern es wurde gewissermaßen von Außen her durch eine Auswahl vorgegeben. Diese Tatsache wird durch die Wertezuweisung im Noponator OP=() artikuliert. Oder anders gesagt, ich hab's mir selber ausgedacht.

Hinweis: OP=() heißt Leeroperator (OP=(): ERROR="no_operation_performed";) weil inhaltlich gilt ()=RA und dies stellt die Vollkommene Leere dar. Und wenn ich mal groß bin, heirate ich den Sarotti-Mohr und kauf' mir vom Geld das ich von ihm bekomme Aktien von Milka. Meine Rosenquarzkugel ist nämlich so lila wie Milka's Kuh. Ich werde in eurer seltsamen Welt noch 'ne Menge Spaß haben!



Ein NOP ist zwar nicht wie eine Liste geordnet, aber beim Generieren des NOPs kann man jedem einzelnen Element über die Systemvariable (OP) einen Operatorindex zuweisen, was die Berechnungsreihenfolge und den Synthesealgorithmus der einzelnen Elemente durch Sub-Turing-Maschinen widerspiegelt.

Wobei n die Anzahl der Operationen des NOP-Generators (Noponator) darstellt, welche notwendig ist um das NOP korrekt zu berechnen. Bevor ein NOP erzeugt werden kann, muß n ein Wert zugewiesen werden. Diese dem mathematischen Laien sicher übertrieben wirkende Forderung ist jedoch unbedingt notwendig um Mengen exakt und widerspruchsfrei definieren zu können.

Bei einer immanenten Operatorsequenzierung muß n im erzeugenden Algorithmus vorhanden oder gegeben sein (Axiom (4) der Guthen Sternwanderey), oder es wird von Außen her, also hier von dem Benutzer angegeben.

Bei iBLUBBER handelt es sich um eine transzendente Syntheseleistung, da diesmal zwar eine Operatorsequenz existiert aber das Zeichen von Außen her vorgegeben wird und kein interner Algorithmus existiert, der das Zeichen auswählt. Ebenso ist n, die Anzahl der Zeichen, bekannt und wird vom User angesagt.

var: Symbol;
NOP=iBLUBBER: 108.[OP=index.(): (Symbol)] =
[
OP=001: (°); OP=002: (!); OP=003: ("); OP=004: (§); OP=005: ($); OP=006: (%);
OP=007: (&); OP=008: (/); OP=009: (); OP=010: ()); OP=011: (=); OP=012: (?);
OP=013: (>); OP=014: (<); OP=015: (|); OP=016: (@); OP=017: (*); OP=018: (+);
OP=019: ('); OP=020: (#); OP=021: (~); OP=022: (_); OP=023: (-); OP=024: (:);
OP=025: (.); OP=026: (;); OP=027: (,); OP=028: ({); OP=029: ([); OP=030: (]);
OP=031: (}); OP=032: (\); OP=033: (€); OP=034: (q); OP=035: (w); OP=036: (e);
OP=037: (r); OP=038: (t); OP=039: (z); OP=040: (u); OP=041: (i); OP=042: (o);
OP=043: (p); OP=044: (ü); OP=045: (a); OP=046: (s); OP=047: (d); OP=048: (f);
OP=049: (g); OP=050: (h); OP=051: (j); OP=052: (k); OP=053: (l); OP=054: (ö);
OP=055: (ä); OP=056: (y); OP=057: (x); OP=058: (c); OP=059: (v); OP=060: (b);
OP=061: (n); OP=062: (m); OP=063: (W); OP=064: (E); OP=065: (R); OP=066: (T);
OP=067: (Z); OP=068: (U); OP=069: (I); OP=070: (O); OP=071: (P); OP=072: (Ü);
OP=073: (A); OP=074: (S); OP=075: (D); OP=076: (F); OP=077: (G); OP=078: (H);
OP=079: (J); OP=080: (K); OP=081: (L); OP=082: (Ö); OP=083: (Ä); OP=084: (Y);
OP=085: (X); OP=086: (C); OP=087: (V); OP=088: (B); OP=089: (N); OP=090: (M);
OP=091: (^); OP=092: (1); OP=093: (2); OP=094: (3); OP=095: (4); OP=096: (5);
OP=097: (6); OP=098: (7); OP=099: (8); OP=100: (9); OP=101: (0); OP=102: (ß);
OP=103: ( ); OP=104: (²); OP=105: (³), OP=106: (´); OP=107: (`); OP=108: (RA);
];

Doch wie schon gesagt, eine Information fällt nicht vom Himmel, denn Gott würfelt ja nicht. Die Reihenfolge der Zeichen die sich nun durch die Operatorsequenz ergibt stellt nun aber grundsätzlich auch wieder eine Information dar und *muß* daher von irgendwo herkommen. Beispielsweise durch die manuelle Eingabe des Users, der durch seine Auswahl eindeutig bestimmt welchem Operatorindex welches Zeichen zugeordnet wird (Transzendente Operatorsequenzierung) oder durch einen Algorithmus innerhalb unseres Systems dessen entsprechende Anwendung und Erzeugung unseres Elementes von diesem Operator repräsentiert wird. (Immanente Operatorsequenzierung)
Der Operatorindex ist durchaus bedeutsam bei Funktionsverkettungen, Prozeduraufrufen und ähnlichem bei dem es auf die Reihenfolge ankommt.
Man könnte also auch hier die einzelnen Elemente in einer anderen Reihenfolge darstellen, für den Magischen Zentral Computer spielt bei der Aufrufabfolge nur der Operatorindex eine Rolle. Bei der Darstellung (Bildschirm, Ausdruck) eines NOPs ordnet der 2MPU normalerweise die operatorindizierten Elemente nach der Operatorsequenz an, also von 1 bis n.
Aber wir können auch ein NOP definieren welches auf einem Algorithmus beruht, der in mehreren Turing-Maschinen abläuft bezeichnet mit OP=index, die zum Ende kommen und damit ein festes Ergebnis t liefern. Damit lassen sich auch leicht unendliche Mengen beliebiger Art erzeugen, denn es gibt da einen ganz fiesen Trick, den erklär ich aber später.

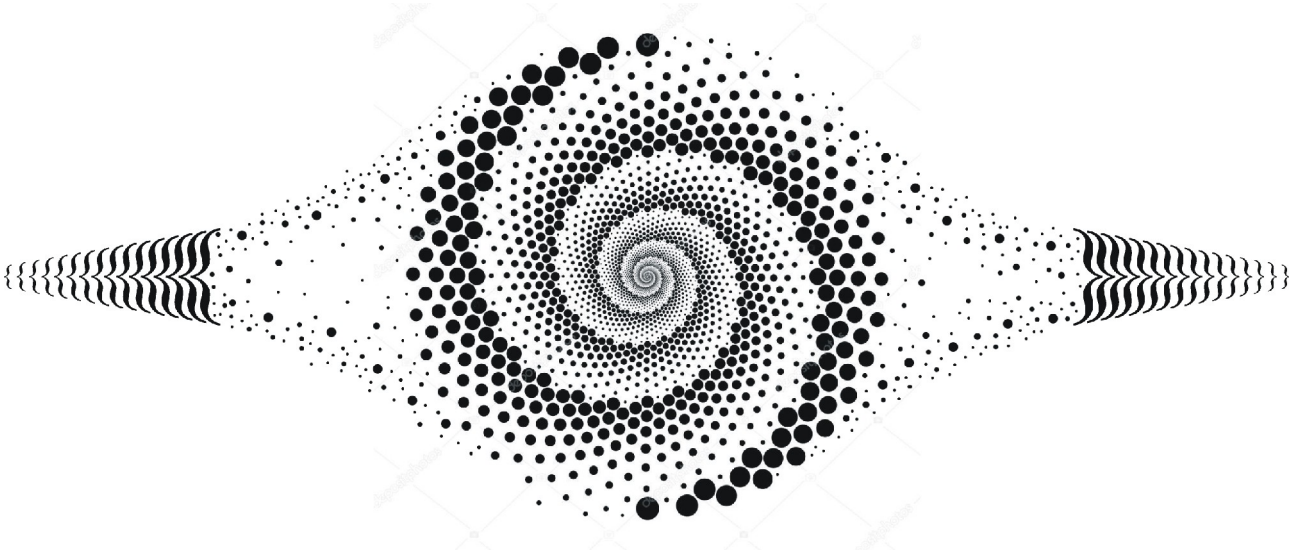


Bild oben: Die größte Ordinalzahl nach Dr. Finsler in graphischer Darstellung. Ob das geht? Niemand weiß es.

```
DEF=next:
[
next := nopFUN*next;
next=RA;
];
```

Hier haben wir die Idee der Iteration selbst kristallisiert. Weder next noch nopFUN haben irgendwelche Werte, brauchen sie auch nicht, es geht eben um die Idee der Iteration.
Der Vorteil dieser Definition liegt in ihrer Universalität, denn je nachdem wie nopFUN und next festgelegt werden erhält man induktive Zahlen, die Aleph-Reihe, Sternwandermengen, unendliche Ordinalzahlen oder vielleicht noch mehr seltsame Dinge.
Wir definieren nun integer über next indem wir der unbestimmten Funktion nopFUN einfach den Wert next zuweisen und für next integer setzen, was zu einer induktiven Definition von integer führt und wir dafür nun einen Startwert benötigen.
Als Startwert wird jetzt einfach die 0 gewählt welche das wohldefinierte Church-Numeral λ fx.x sei. Alle next*n seien ebenfalls die entsprechenden Church-Numerale womit dieses zunächst unbestimmte next* durch die Definition von integer eine Bedeutung gewinnt, nämlich die Idee der jetzt konkret werdenden Iteration als Syntheseleistung einer Objektfolge.

```
DEF=integer:
[
nopFUN=integer: next*integer;
integer=0;
];
```

Integer ist induktiv definiert. Erfolgt keine Anwendung der Definitionszeile und damit auch keine Funktionsanwendung, hat Integer bei dieser Nichtanwendung den Wert 0.
Die 0 steht für keine Anwendung von Integer: 0.integer; Da ja 0 ein Church-Numeral sei und damit eine Funktion ist, die auf Integer angewendet nichts bewirkt, weil keine Anwendung stattfindet, dürfen wir mit Fug und Recht 0.integer schreiben.
Danach erfolgt aufgrund der induktiven Definition der nächste Induktionsschritt, repräsentiert durch 1.Integer bzw. allgemein: n.integer
Wir können nun alle Zahlen als Idee der Iteration in entsprechender Gradation definieren, und zwar identisch zu den Church-Numeralen. Die Funktion next in ihrer nicht induktiven Anwendung wird erst durch die Wertezuweisung nopFUN=next erzeugt und gewissermaßen damit entleert, denn in der Definition von next kommt ja nopFUN vor, womit ein fortlaufender Substitutionsprozeß eingeleitet wird, der nopFUN im Unendlichen tatsächlich zum Verschwinden bringt.
Man denke an die Definition der unfundierten Menge omset := {omset};
Was ist omset letzten Endes? Es gibt aber kein Ende, denn der Substitutionsprozeß um omset "dingfest" zu machen oder "auszurechnen" kann nicht enden und ist damit nicht existent, wenn wir an einen Algorithmus den Anspruch stellen einmal fertig zu werden und ein Ergebnis zu liefern. Der einzige Ausweg aus diesem Paradoxon ist die Transzendierung des "Prozesses ad infinitum" selbst zu einer neuen definierten Funktion.
Dies entspricht dann ganz klar dem aktual Unendlichen, während das potentiell Unendliche in der Immanenz des Prozesses selbst, also in sich eingeschlossen zurück bleibt.

Nun ist integer eine induktiv definierte Funktion mit dem Church-Numeral 0 als Startwert und keine Zahl im Sinne eines Church-Numerals, welches natürlich auch nur eine Funktion ist. Wir dürfen schreiben: 0.integer=0; denn keine Anwendung von integer liefert den Startwert von integer, nämlich die 0. Wird integer einmal angewendet entsteht 1.integer=next*0;
Oder fortlaufend:

0.integer = 0;
1.integer = next*0;
2.integer = next*(next*0);
3.integer = next*(next*(next*0));
.....

Die n-fache Anwendung impliziert: n.integer = n.next*0;
Denn die n-fach verschachtelte Anwendung von next können wir zu n.next zusammenfassen und auf die 0 anwenden. Nun müssen wir nur noch die Funktion n.next*0 definieren, denn noch ist nicht einmal die Anwendung next*0 ein konkretes mathematisches Objekt , es existiert lediglich eine induktive Definition von integer auf Basis einer wiederum induktiv beschriebenen Funktion next und dem netten Startelement RA durch das eine größtmögliche Abstraktion von next erreicht wird. Daher können wir gar nicht wissen was next*0 oder n.next*0 überhaupt für ein Objekt darstellen soll. Gerade diese aus der Abstraktion folgende Unbestimmtheit erlaubt uns aber n.next*0 wie gefordert zu definieren ohne die Funktion next festlegen zu müssen!

Wir legen n.next*0 fest:

```
DEF=n.next*0:
[
var: n;
n.next*0 := n;
];
```

Die Funktion integer liefert nun bei n-facher Anwendung das entsprechende Church-Numeral und so erzeugt integer brav und voller Freude alle natürlichen Zahlen welche sich aus diesen Numeralen ergeben:

n.integer = n.next*0 = n;

Nun können wir auch von next*k annehmen, daß es sich um ein Church-Numeral handelt, wenn auch k selbst eines ist:

next*p = next*(p.next*0) = q.next*0 = q;

Und für next*0 muß gelten:

next*0 = next*(0.next*0) = 1.next*0 = 1;

Wie sich leicht erkennen läßt, wenn wir die Addition voraussetzen, besteht zwischen q und p die Beziehung q = p+1;
Daher können wir die Funktion succ im Sinne des Lambda-Kalküls mit next*n identifizieren, solange next auf ein Argument angewendet wird (next*k) und dieses vom Typ integer ist. Die Funktion next selbst jedoch ist weit allgemeiner gefaßt und hängt von der Nachfolgerfunktion nopFUN und dem Startwert ab. Bei Church-Numeralen ist aufgrund ihrer Lambda-Definition diese Funktion und ihr Argument irrelevant, da sie im Abstraktionsteil stehen.
Wir können über das induktiv und abstrakt definierte next jedoch genauso gut völlig neue Konstrukte erzeugen, die von der Struktur dieser Funktion und der Eigenschaft ihres Startwertes abhängig sind. Das garantiert prickelnde Überraschungen von denen alle Logiker Nachts träumen!

Die Funktion succ ist die klassische Nachfolgerfunktion der induktiven Zahlen wie schon im Lambda-Kalkül definiert: succ := λ nfx.f(nfx);
Das Argument von succ muß natürlich ein Church-Numeral also eine induktive Zahl sein, da die Funktion next in seiner nicht induktiven Anwendung (next*) aufgrund der Definition von integer daran gebunden ist.

succ*0 = next*0 = 1;
succ*1 = next*1 = 2;
succ*2 = next*2 = 3;
.....
succ*n = next*n = n+1;

Addition: k+n := k.succ*n;

Integer ist nun weder die Menge der Natürlichen Zahlen, noch ist Integer selbst eine induktive Zahl, sondern eher die Idee des Iterieren's ad infinitum und dient daher auch als Ausgang zur Definition der Unendlichkeit.
Metapher: Der erste von Hand angestoßene Wink einer Winkekatze ist die 0. Dann folgen weiter Winks, und immer noch ein Nachfolgewink. Die Winkekatze selbst ist aber weder ein Wink noch die Menge aller Winks sondern steht für das nicht endende Winken als Objekt, wobei das Winken selbst in den Hintergrund tritt und die unendliche Wiederholung zum Wesentlichen wird, da die Wiederholung den Vorgang des Winken's entleert und ihn damit zum Verschwinden bringt. Versuchen Sie es mal, kaufen Sie sich eine kleine Winkekatze und beobachten Sie diese dann beim Winken! Eine viertel Stunde reicht in der Regel für Einsteiger um ein Gefühl von Unendlichkeit zu erzeugen.
Achtung! Dieses Verfahren ist jedoch nicht geeignet für psychisch divergente Personen. Dosieren Sie unter Drogeneinfluß die Winkekatze sparsam! Nicht im Spiegel beobachten! Aber ansonsten kann man sich damit Aleph Null tatsächlich ins Wohnzimmer holen.

Der index ist die Unruh in der Uhr, der Geist in der Maschine, das Zählwerk, die TicTac-Energie. Geeignet für Sternwandermengen bis zur Gradation 0.aleph.

```
DEF=index:
[
nopFUN=index: index + 1;
index=1;
];
```

0.index = 1;
1.index = 2;
2.index = 3;
.....
n.index = n+1;

Wir definieren nun das aleph über next indem wir für die freie Funktion nopFUN einfach die Potenzfunktion 2 hoch x setzen und als Startwert x=aleph wählen, was zu einer induktiven Definition von aleph führt und wir wiederum einen Startwert benötigen für den wir integer einsetzen. Die Potenzfunktion führen wir aus durch iterieren der Multiplikation 2*1 und der Startwert integer erweist sich hier nun bei der Nichtanwendung von aleph als 0.aleph, was eine Entsprechung in der Kardinalzahl Aleph-null hat. Nun haben wir eine Progression von abgestuften Unendlichkeiten, die sich den natürlichen Zahlen nicht unähnlich verhalten.

```
DEF=aleph:
[
nopFUN=aleph: aleph.2*1;
aleph=integer;
];
```

0.aleph = integer;
1.aleph = integer.2*1;
2.aleph = (integer.2*1).2*1;
3.aleph = (integer.2*1).((integer.2*1).2*1);
.....
(n+1).aleph = n.aleph.2*1;

Da n.aleph eine etwas sperrige Symbolisierung für eine Funktion ist, und man für ein Church-Numeral ja auch nicht n.integer sondern einfach n schreibt, ist im 2MPU folgende Schreibweise festgelegt:

```
DEF=(°):
[
var: n;
n° :=  n.aleph;
];
```

So kann man bequem mit den Unendlichkeiten umgehen und Gradationen angeben wie $1°.X oder $0°.X Man darf natürlich auch schreiben: $(1.aleph).X oder $(0.aleph).X

```
DEF=hyper:
[
nopFUN=hyper: hyper*hyper
];
```

hyper ist ein Transzendator. *(So ein Fubbes, aber Scooter gefällt das bestimmt. Anm. des Übersetzers)*

Anwendung:
hyper läßt sich aufgrund seiner Definition nur anwenden, wenn ein Induktionsstart in Form einer Funktion hyper als Wert zugewiesen wird. Dies ist erforderlich, da hyper induktiv als unbestimmte Funktion definiert ist. hyper kann nur über einen Iterator (Church-Numeral) angesteuert werden. Sei T einfach eine beliebige Funktion, die auf sich selbst anwendbar ist.

hyper=T:
0.hyper = T;
1.hyper = T*T;
2.hyper = (T*T)*(T*T);
3.hyper = ((T*T)*(T*T))*((T*T)*(T*T));
.....proceed to 0.aleph.....
(hehe, Omega-Kombinator neu erfunden oder was? Anm. des Übersetzers)

Die bekannte Lambdafunktion λx.xx repräsentiert tatsächlich die hier definierte Funktion hyper. Wenn diese Lambdafunktion auf die Leere Menge angewendet wird, natürlich unter der Voraussetzung wie wir eine Menge als Funktion auffassen, wird diese Funktion auf sich selber angewendet, woraus durch Iteration die von Neumann-Zahlen entstehen. Da die Menge nur als solche definiert wurde, müssen wir sie nun aber noch als Funktion definieren:

```
DEF=nopFUN:
[
nopFUN=set: set*object := set <+ object;
];
```

Hinweis der Redaktion: Hier wurde eine Funktion in NOPTONIC definiert. Man könnte diesen Zusammenhang aber auch unter alleiniger Verwendung der zwei Relatoren [=] und [:<] definieren: [set*object] => [object :< set];

Hier ein Beispiel für die Anwendung von hyper mit Mengen als Funktion:
hyper=zset:
3.hyper =
(({ }*{ })*({ }*{ }))*(({ }*{ })*({ }*{ })) =
({{ }}*{{ }})*({{ }}*{{ }}) =
{{ }, {{ }}}*{{ }, {{ }}} =
{{ }, {{ }}, {{ }, {{ }}}};
(Mmmmmmh eszet! Das waren doch die lecker Schokoladeplättchen von Staengel & Ziller, nicht aber von Zermelo & Fraenkel. Die hatten damit nichts zu tun. Anm. der Redaktion)

Neumann-Zahlen sehr schön definiert. Oder anderes Mengengeblubber. Kommt auf den Startwert an. In Λ ausgedrückt müßte das für eine beliebige Funktion in NOPTONIC so aussehen:
hyper=nopFUN: n.hyper = [[]:type=Λ ; n(λx.xx)nopFUN];
wobei n ein Church-Numeral sei und nopFUN eine beliebige Funktion die auf sich selbst anwendbar ist. zset sei natürlich die im gesamten NOPTONIC so benannte Leere Menge: zset := { }; Man könnte aber auch Sternwander selbst verwenden:

hyper=$.X-BIGBANG: aleph.hyper = ?;
(Menno, jetzt aber nicht gleich übertreiben! Anm. der Redaktion im Namen des Herrn)

Wir wollen nun versuchen uns dem RA-Genius zu nähern. Nach langjähriger Überlegung komme ich zu dem Schluß, daß dieses RA auf jeden Fall folgende Eigenschaften aufweisen sollte:

RA:type=[zset*RA := { }; RA*object = object;];

Was bedeutet das? Ein Objekt welches gar nicht vorhanden ist? Ist das etwa ein "Unding", also überhaupt gar kein richtiges, echtes Objekt, welches in unserem System NOPTONIC "existiert"? In der Tat, hier handelt es sich offenbar um das "Nichts". Wenn wir das Nichts wie eine Funktion anwenden, kommt immer das Funktionsargument heraus, denn die Anwendung des Nichts ist gleich mit keiner Anwendung, also die Identitätsfunktion!
Sehr schön artikuliert im Lambdakalkül mit id := λx.x was ja der Identitätsfunktion entspricht, denn dieser Lambda-Ausdruck bewirkt in seiner Anwendung auf ein Objekt nichts. Heraus kommt das Objekt selbst, unverändert. Aber id ist ein Etwas im Sinne eines Objektes unserer Auffassung, nämlich die Artikulation der Identitätsfunktion und damit gilt: zset*id = {id} ={λx.x}; id*id = id;
Aber es gibt eine Funktion als Lambda-Ausdruck, die in ihrer Anwendung dasselbe wie id bewirkt: 0 := (λfx.x); 0*g*z = z; *(Achtung! Applikationen sind in NOPTONIC linksradikal, äh, linksassoziativ!)*
Das Church-Numeral Null angewendet auf jede beliebige Funktion ergibt eine neue Funktion, die wie die Identitätsfunktion wirkt, indem sie eben nicht wirkt, da sie aufgrund der Wirkung des Church-Numerals Null nicht angewendet werden soll! Die Frage ist nun, läßt sich dieser Lambda-Ausdruck in id überführen? Beide Funktionen sind in ihrer Wirkung absolut identisch, nur id wird angewendet, während der Ausdruck 0*g in seiner Anwendung keine Anwendung darstellt! Ist 0*g eine Art Funktionssingularität? Denn wie g auch konstruiert ist, da g ja nicht angewendet werden soll, ist g vollkommen irrelevant und wir erhalten wie bei id immer das unveränderte Argument als Ausgabe zurück. Gilt also, ganz egal was g für eine Funktion ist: 0*g = id = λx.x; id ist ein existierendes mathematisches Objekt, nämlich eine Funktion, man kann es mit anderen Lambda-Ausdrücken durchaus vergleichen weil es selbst auch ein korrekter Lambda-Ausdruck ist. RA als Funktion die ja gar nicht vorhanden ist wirkt intuitiv gesehen zunächst wie id. Doch sobald wir RA als Funktion zulassen geben wir RA eine Vergleichbarkeit und damit eine Objekthaftigkeit. RA als Funktion wäre nur denkbar, wenn es sich anders als id verhält. Einfachste Lösung dafür ist die sogenannte Diskordianische Selbstreferenz, hier als Definition in NOPTONIC:

```
DEF=nopFUN:
        [
        nopFUN=RA:
                [
                RA*object := object;
                RA*RA := { };
                ]
        ];
```

Daraus folgen natürlich die zwei geforderten Gleichungen die RA erfüllen muß:
(1) RA*object = object;
(2) RA*RA = { };

Da Ra kein Objekt sein soll, können wir diese Definition setzen. Wäre RA objekthaft müßte nach Gleichung (1) gelten: RA*RA = RA; Daraus würde aber nach (2) folgen RA = { }; und damit auch RA*RA = {{ }}; was man sicher als Widerspruch auffassen kann. Diese 2 Gleichungen sind nur dann haltbar, wenn RA eine Art Unding ist, ein Etwas, das nicht mit sich selbst identisch sein kann, denn ist es mit sich selbst identisch, so wäre es ja nach der Theorie der Objekte selbst ein Objekt und unsere geforderten Gleichungen eine Contradictio und somit hinfällig.
RA kann also mit Objekten überhaupt nicht in eine Relation gesetzt werden, wohl aber mit sich selber! Die Menge der Objekte jedoch die nicht mit sich selbst identisch sind ist nach dem System von Zermelo & Fraenkel die Leere Menge. Die Diskordianische Selbstreferenz von RA impliziert die Forderung: #[RA=RA] => [RA#RA]; denn keine Funktion ist in der Lage diese beiden Gleichungen zu erfüllen, wenn diese Funktion selbst ein Objekt ist.

Daher fordern wir also von RA diese Eigenschaft der Nicht-Identität und es ergibt sich zwanglos nach Z & F: zset*e:type=[e#e] = { }; => zset*RA = { }; und erhalten damit aus unserer Definition der Diskordianischen Selbstreferenz eine weitere Gleichung mit RA. Wir fassen diese zusammen zur Heiligen Dreifaltigkeit des RA:

```
DEF=magicRA:
        [
        DEC=magicRA:
                [
                OP=3: nopFUN*RA = nopFUN;
                OP=1: RA*object = object;
                OP=2: RA*RA = { };
                ]
        ];
```

Damit haben wir in der Tat das Unding RA dingfest gemacht und können sogar einige Angaben über sein Verhalten machen ohne daß dieses Unding überhaupt existiert. Halleluja!
Als Definition sieht das so aus:

```
DEF=RA:
[
RA := RA:type=[RA#RA]
];
```

Es scheint als wäre RA so etwas wie die abstrakte Idee der Idee ..... des Objektes. Was aber soll das nur sein? Niemand weiß es. Und vielleicht werden wir es nie, niemals nie, erfahren..... Wenn Sie es dennoch unbedingt wissen wollen, fragen Sie doch einfach die Rosenquarzkugel! Es sind folgende Mengen in NOPTONIC global definiert:

```
DEF=Alpha&Omega:
[
set=zset: { };
set=omset: {omset};
];
```

Und es gilt, wie jedermann sicher leicht nachvollziehen kann:

- Alle Objekte sind Fixpunkt von id und RA.
- omset ist Fixpunkt von zset, denn: zset*omset = omset;
- omset ist Hyper-Fixpunkt, also Fixpunkt von sich selber: omset*omset = omset;
- id ist Hyper-Fixpunkt: id*id = id;

Nun läßt sich aber auch das Objekt an sich definieren, nämlich die Dinge welche mit sich selbst identisch sind was wir für alle Dinge außer dem Unding nach der Theorie der Objekte fordern:

```
DEF=object:
[
object := object:type=[object=object]
];
```

Tritt ein, sei Gott!

Da wir nun das Ding an sich als Objekt definiert haben, können wir tatsächlich die Definition selbst, welche nach der Theorie der Objekte ja auch ein Ding sein muß, definieren.
Wow! Erklärt das mal der Spanischen Inquisition, die hab ich nämlich jetzt noch nicht erwartet. Die Definition an sich oder auch die Idee der Definition wird jetzt zur Bewußtwerdung der totalen Freiheit, die Möglichkeit einfach alles zu definieren, oder auch der freie Wille, so frei, daß einem Angst und Bange wird.

```
DEF=DEF:
[
DEF := DEF
];
```

Das ist das Schöne an NOPTONIC, man kann sich alles selber definieren und damit unglaublichen Unfug anrichten. NOPTONIC ist also gefährlicher wie Religion. Nur man weiß halt immer, daß es die eigenen Definitionen sind, weil man könnte sie ja auch ändern oder gleich so anlegen, daß ihnen Werte von Außen zugewiesen werden. Daher ist NOPTONIC jeder Religion überlegen, denn die "Außenwelt" (operative Zugänglichkeit) bleibt in NOPTONIC erhalten. Psychiater nennen das Krankheitseinsicht. HöHö, NOPTONIC als Krankheit mit Einsicht in NOPTONIC...
Quasi virtueller Wahnsinn oder sagen wir mal nur eingebildeter Wahnsinn. Auch nicht schlecht.
(Wahnsinn als Wahnidee ist ja jetzt echt wie der Omega-Kombinator im Kopf oder einfach nur Wahnsinn²! Aber bewußt eingebildeter Wahnsinn, das entspräche dem Fixpunkt-Kombinator. Anm. des Übersetzers)
Die Prinzessin: Das ist ja wie Ficken, Kunst & Kotzen in einem!

Funktionen und Applikationen

NOPTONIC-Funktionen nopFUN können benannt global oder lokal in einem Programmablauf definiert werden. Auch ist es sinnvoll sich bewährte und gebräuchliche Standardfunktionen wie arithmetische Funktionen oder oft verwendete algorithmische Standards wie Potenzmenge oder POWSET, Loops und diverse Sternwander (X-BIGBANG) zu definieren. Die globale Definition wird mit DEF=keyWORD eingeleitet und ist somit immer benannt. Die individuelle globale Definition wird mit DEF=nopFUN und anschließender Benennung eingeleitet. Lokale Definitionen innerhalb eines Programmablaufes sind ebenfalls möglich, bleiben jedoch auf ihre Wirkebene und die darunterliegenden begrenzt.
Durch Church-Numerale wird über den Subtextor (.) eine einfache und übersichtliche Iteration erreicht. Mit dem Parameter "NOP-Sprachartikulation" ([]:type) kann in einem NOP eine Funktion in jeder vorher in der Sprachbibliothek von NOPTONIC abgelegten Programmiersprache artikuliert werden. Die Sukzessorfunktion wird hier z. B. in Λ (Lambda-Kalkül) ausgedrückt:

```
nopFUN=succ:
[
[]:type=Λ ; succ := λn.λf.λx.f(nfx);
];
```

Jetzt bewirken die Funktionen, die mit dem Universaltextor (*) übertragen werden:
succ*n = n+1 oder iterativ angewendet: k.succ*n = n+k
Definitionsbereich sind standardisiert Church-Numerale, aber es können auch an Operatorindizes gebundene beliebige NOPs nachfolgen, wenn dies in der Parameterinformation der Funktion angegeben wird (arithmetische Listenbearbeitung).
Das kann für sehr große NOPs oder bei einer Gödel-Numerierung überaus praktisch sein. Alle Funktionen werden durch den Universaltextor (*) oder den Subtextor (.) übertragen. NOPTONIC ist linksassoziativ. Der Universaltextor wirkt auf das gesamte NOP als Argument. Doch Vorsicht! Der Subtextor wirkt dagegen distributiv falls das Argument ein NOP mit mehreren Objekten ist. Es entsteht dann ein neues NOP mit neuen von der Funktion erzeugten operatorindizierten Objekten.
Weiter läßt sich eine Wahrheitsfunktion REAL über nopFUN definieren und mit einer positiven Logik besetzen. Argumente sind Aussagen (DEC), die Funktionswerte seien entweder 1 oder 0. Was bedeutet 1 steht für den Wahrheitswert “wahr” und 0 für den Wahrheitswert “falsch”. Später werden wir auch Werte zwischen 0 und 1 einführen, also Wahrscheinlichkeitswerte für einen Wahrheitswert (Fuzzy-Logic). Das eröffnet uns die Möglichkeit einer Ankoppelung an die Systeme der Quantenmechanik.
Doch zunächst sei der Wahrheitswert unseres Aussageoperators ein Element aus:
NOP=BOOL: [OP=1.BOOL: 0; OP=2.BOOL: 1;];
Dieser Ausdruck ist gleichwertig mit der klassischen Mengenschreibweise setBOOL := {0, 1}; Jedoch verwendet man die klassische Mengenschreibweise bevorzugt für die Matrixmenge eines Sternwanders oder dessen Definitionsmengen.
Nun läßt sich aus Grundsatz (1) folgende Wahrheitswerte des Aussageoperators ableiten: REAL*[$.X*p :< $.X] = 1;
Da jeder beliebige Stern aus einer Matrixmenge des Grades $ nach Grundsatz (1) der Guthen Sternwanderey ein Element der zugehörigen Matrixmenge $.X ist.
Oder auch: REAL*[{ } < $.X] = 1; Da die leere Menge Teilmenge jeder Menge ist.
REAL*[DEC := REAL*DEC :< BOOL] = 1; (Achtung wirkt selbstbezüglich! Geht aber.)
oder iterativ formuliert: REAL*[DEC := [DEC := [REAL*DEC :< BOOL]]] = 1; was gleichwertig durch Einsetzen in sich selber ist mit: REAL*[DEC := [DEC := [DEC := [REAL*DEC :< BOOL]]]];
Das heißt, die Aussage, daß diese Aussage, daß diese Aussage, daß diese Aussage....., daß diese Aussage entweder wahr oder falsch ist, ist wahr. Jetzt haben wir die Unendlichkeit in eine kleine Murmel eingefangen. Durch eine solche unendliche Iteration der Wahrheitsfunktion werden Fundamental-Aussagen möglich und man kann auch durch endliche Iteration Falschaussagen bewußt artikulieren und damit das Wissen um das Nichtwissen als reinen Informationsgehalt einer Aussage dingfest machen. Wie läßt sich durch Algorithmen die Lüge von der Wahrheit trennen oder was soll das sein, diese Wahrheit? Ein eher amüsant bekiffter Gedanke zum Halteproblem oder als der Herr HerrGott das Lügen lernte...

Mit den logischen Operatoren und dem Conclusionsoperator lassen sich einfache Programme mit entsprechenden Kontrollstrukturen verwirklichen. Beispielsweise eine Abfrage mit einer entsprechenden Reaktion:

```
app=LOOP:
[
OP=1.LOOP:
var: u=(), k=(), c=();
OP=2.LOOP:
[
(u<k) % (u=k) => c := u+c;
(u>k) => [OP=1: WRTLN="LOOP is ready! Summe="c; OP=2: OP=0.LOOP;];
];
OP=3.LOOP:
u := u+1;
OP=4.LOOP: OP=2.LOOP;
];
```

Operation 1:
Definition und Wertezuweisung der Variablen, hier muß der Benutzer beim Programmstart einen Wert eingeben, man könnte aber auch Werte von parallel laufenden Programmen übernehmen.

Operation 2:
Wenn u kleiner oder gleich k, so folgt:
Neuer Wert von c ergibt sich aus dem aktuellen Wert von u und dem aktuellen Wert von c;
Wenn u>k so folgt:
Führe den Befehl WRTLN="LOOP is ready! c="c aus.
Auf einem Ausgabegerät (Bildschirm, Drucker) erscheint: LOOP is ready! Summe=()
Führe OP=0.LOOP aus, was eine Operatorübergabe zurück in die Laufzeitumgebung von LOOP darstellt, worauf OP=0.LOOP ausgeführt wird: OP=0.LOOP: OP=() da jede Nichtanwendung jedes beliebigen Programmes durch den Leeroperator artikuliert wird. Da 0 kein index ist, kann auch keine Operation nachfolgen was zu einem "Verschwinden" der Laufzeitumgebung und dem Anhalten von LOOP führt.

Operation 3:
Erhöhe der Wert von u um eins.

Operation 4:
Die Operation 4 besteht daraus Operation 2 auszuführen. Dies ist eine Operatorübergabe und führt dazu, daß die Operatorsequenz von exeLOOP, also die Laufzeitumgebung von LOOP, erneut LOOP ausführt und bei OP=2.LOOP beginnt. Die Werte von u, k und c sind ja bereits vorhanden, daher können wir uns OP=1.LOOP sparen. Theoretisch könnte man aber auch schreiben: OP=4.LOOP: OP=LOOP, dann wird exeLOOP weiter sukzessiv ausgeführt und beginnt wieder bei OP=1.LOOP

Was macht dieses Programm eigentlich? Für die Belegung der Variablen u=1 und c=0 berechnet es die Summe der Zahlen von 1 bis k und gibt das Ergebnis unter der Variablen c aus.
Dies soll natürlich nicht praxisorientierte Programmierungsanwendungen dienen, sondern hier geht es zu allererst darum, einfache Sternwander wohldefiniert zu formulieren. Um komplexere und vor allem sehr große Noptronische Mengen zu erzeugen und ihre Eigenschaften zu untersuchen müssen diese in einem Großrechner unter Verwendung einer geeigneten Programmiersprache formuliert werden. Oder man implementiert NOPTONIC unter Verwendung eines Hodge*Podge-Generators in eine kompatible Sprache. Dafür wäre dann aber eine noptronische CPU notwendig, die ja dann selbst durch eine Noptronische Menge, welche in einem handelsüblichen Prozessorkern als virtuelle Maschine emuliert werden könnte. Eine elegantere Lösung wäre natürlich ein Quantenprozessorkern. Entweder als photonische Matrix codiert in einer Rosenquarzkugel oder aus mehrfach verschränkten Neutrinos in einem geschichtetem zweidimensionalen Neutronenkristall. Fragen Sie bei einem UFO-Kontakt einfach die Außerirdischen!

Ausführung:

Allgemeine Form der Ausführung einer app in NOPTONIC:
Das miniPOOPI exe generiert hier eine Laufzeitumgebung um die Applikation LOOP:

```
exeLOOP := ().[OP=index.exeLOOP: ().[OP=index.LOOP];];
```

Aus der Operatorübergabe OP=4.LOOP: OP=2.LOOP ist ersichtlich, daß die Operationen von LOOP mehrfach angewendet werden sollen. Daher ist eine Laufzeitumgebung sinnvoll, denn sie ermöglicht ein Springen von einem Operatorindex zu einem anderen durch eine Operatorübergabe.
Gäbe es keine Operatorübergabe von OP=4.LOOP: OP=2.LOOP würde aufgrund des index LOOP wieder bei OP=1.LOOP beginnen was wir uns aber sparen können.

```
exeLOOP := ().[OP=index.exeLOOP: ().[OP=index.LOOP]];
=>
exeLOOP=
[
OP=1.exeLOOP:
[
OP=1.LOOP: u=1; k=4; c=0;
OP=2.LOOP: (1<4) % (1=4) => c = 1+0;
OP=3.LOOP: u = 1+1;
OP=4.LOOP: OP=2.LOOP;
];

OP=2.exeLOOP:
[
OP=2.LOOP: (2<4) % (2=4) => c = 2+1;
OP=3.LOOP: u = 2+1;
OP=4.LOOP: OP=2.LOOP;
];

OP=3.exeLOOP:
[
OP=2.LOOP: (3<4) % (3=4) => c = 3+3;
OP=3.LOOP: u = 3+1;
OP=4.LOOP: OP=2.LOOP;
];

OP=4.exeLOOP:
[
OP=2.LOOP: (4<4) % (4=4) => c = 4+6;
OP=3.LOOP: u = 4+1;
OP=4.LOOP: OP=2.LOOP;
];

OP=5.exeLOOP:
[
OP=2.LOOP: (5>4) => [OP=1: WRTLN="LOOP is ready! Summe="10; OP=2: OP=0.LOOP;];
];

OP=6.exeLOOP:
[
OP=0.LOOP: OP=()
];
]
```

Wünscht man z. B. für den Fall u>k eine andere Weiterführung und keinen Stop, bräuchte man doch nur das entsprechende Programm anstatt OP=0.LOOP einzutragen. Zum Beispiel:
OP=2: OP=HOLLEBOLLE()
Das geht natürlich, aber entweder es existiert bereits eine Umgebung von HOLLEBOLLE, oder man muß durch Variablendefinition oder Variablenübergabe im Parameterteil eine erzeugen. Existiert bereits eine Umgebung kann man auf den Parameterteil verzichten und HOLLEBOLLE startet automatisch wenn kein OP-index angegeben wird bei OP=1.HOLLEBOLLE mit der nächsten index-Operation:

```
OP=(max.index+1).exeHOLLEBOLLE:
[
OP=1.HOLLEBOLLE: [.....];
OP=2.HOLLEBOLLE: [.....];
.................................
];
```

Man könnte natürlich auch bei einem beliebigen Operator von HOLLEBOLLE beginnen. Wenn nun in HOLLEBOLLE wieder eine Operatorübergabe stattfindet z.B. OP: OP=2.LOOP; oder OP=99.HOLLEBOLLE: OP=2.LOOP; wird LOOP wieder fortgesetzt.
Da können natürlich auch schnell Inkonsistenzen entstehen, die Syntax erlaubt so manches, daher gibt es die Forderungen von der Konsistenz der Laufzeitumgebung:

Jedes Programm hat nur genau <u>eine</u> Laufzeitumgebung.

Wie Sie sehen, lassen sich schon mit den einfachsten Mitteln beliebig verschachtelbare, schön blubbernde und wild wuchernde Programme erzeugen von denen niemand weiß was sie so treiben.
Wir können nun eine Standard-app in allgemeiner Form definieren:
Nennen wir sie LOOP weil sie von u nach k zählt und dabei jedesmal das Unterprogramm X ausführt. Dabei kann X selbst eine Variable vom Typ app, also auch ein externes Programm sein.

```
app=LOOP:

[
OP=1.LOOP:
var: u=(), k=(), X=();

OP=2.LOOP:
[
(u<k) % (u=k) => X;
(u>k) => [OP=1: WRTLN="LOOP is ready!"; OP=2: OP=0.LOOP;];
];

OP=3.LOOP:
u := u+1;

OP=4.LOOP: OP=2.LOOP;
]
```

Jetzt haben wir ein kleines Standardprogramm oder ein Modul Namens LOOP das so lange X anwendet oder aufruft bis u>k und dann anhält, aber vorher noch schnell sagt, daß es jetzt fertig ist.
Selbstverständlich funktioniert das nur, wenn das verlinkte Programm X über eine bestimmte Operatorübergabe wieder in die Laufzeitumgebung von LOOP zurückverweist. Der Vorteil liegt darin, egal wie tief man in irgendwelchen Unterprogrammen vergraben ist, man kehrt mit dieser Operatorübergabe wieder in exeLOOP zurück.
So kann man alle möglichen Programme miteinander Verbinden wie eine Binde, die alles verbindet und das Ganze wurde lediglich mit exeLOOP gestartet.

All die bekloppten Neologismen von Dr. Rosenquarz offenbaren nun ihren tieferen Sinn, handelt es sich doch hier um komplexe Anwendungen des Magischen Zentralcomputers von Cocos*Palam. NOPTONIC ermöglicht eine Koppelung von mystischen Weltsystemen mit herkömmlichen Computer-Technologie bekannt aus der Werbung in Funk und Fernsehen. Eines Tages wird es möglich sein Teile von Cocos*Palam als **complexREAlity** oder sogar als **nopusDEI** zu spielen!
Diese Spiele wurden speziell entwickelt um neue kulturrelevante Eingriffe und gute neue Stimulation freizusetzen. Eine Spiel im Spiel im Spiel... Also die Matrix in der Matrix in der Matrix... Und wo ist man jetzt, wenn man erwacht? Wieder in der Matrix? Ist die Matrix nur in sich selber? Wann ist Erntezeit? Ab 10 Milliarden Menschen ist Ernte, sagt die Gritli. Die muß es ja wissen.

Was ist Existenz? Niemand weiß es und die Einzigen die uns nicht weiterhelfen können sind die immer Zigarre rauchenden Philosophen in ihren schwarzen Rollkragenpullis und dem schrägen Blick aus Sonnenbrille, möglichst klein und schwarz verspiegelt, oder dicke Rauschebärte in denen Aschereste beim Reden herum rieseln. Sowas hilft uns wirklich nicht. Besser sind da die Physiker mit ihren immer und ewig unlösbaren Differentialgleichungen für die man aber nur Näherungsgleichungen von Näherungsgleichungen von noch immer sehr, sehr komplizierten unlösbaren Näherungsgleichungen angeben kann.
Der Herr Lesch hat im Fernseher gesagt: "Wir bestehen zu 90% aus Bindungsenergie." Da kam mir eine Idee: Was wäre wenn einfach alles nur aus Bindungen, oder sagen wir mal zu 100% einfach nur aus reinen Beziehungen bestünden täte? (Achtung, Konjunktivpsychose!) Jedes Objekt in unserem gedachten Universum hätte gar keine eigene unabhängige Identität, sondern besteht komplett aus einem Beziehungsgefüge anderer Objekte. Am Ende gibt es dann gar nichts richtig Handfestes! Wo führt uns das hin, wenn wir Objekte immer weiter in neue Beziehungen von Objekten aufbrechen, die wiederum zerlegt werden können? Nehmen wir einfach mal an dieser Prozeß ließe sich ad Infinitum fortsetzen. Schlimmer noch, es ließen sich ja auch zyklisch definierte Objekte erzeugen!
Geht es Ihnen jetzt auch so? Irgendwie beschleicht einem dabei ein recht unwirkliches Gefühl, als hätten solche Objekte überhaupt gar keine Substanz mehr, das Ganze wirkt eher traumhaft fluffig wie eine Art existentielle Zuckerwatte.
Aber wie begann das alles, wenn wir ein solch komplett unfundiertes Universum aufstellen? Wäre es nicht schön von einem Anfang aus nach simplen Regeln, die auf einem Bierdeckel Platz fänden, ein sich stetig ausdehnendes dynamisches Universum abzuleiten?
Man bräuchte nur eine Art Ur-Ding und ein paar Regeln sowie Forderungen, die unsere Objekte erfüllen sollten.
Noch raffinierter ist die Idee, daß ein solches Ur-Ding sogar die Axiome selbst darstellt! Doch gemach. Zunächst ein paar grundlegende Ideen was für "wirkliche Objekte" gelten soll: Wie das Wort schon durchscheinen läßt, geht es hier um Wirklichkeit im Sinne von Wirkungen, also ein Netzwerk kausaler Konstruktion und Bezugnahme der Objekte aufeinander als Grundlage objektiver Realität. Man könnte auch sagen, Wirklichkeit ist, wenn alle gemeinsam objektiv halluzinieren. Klingt doch ganz nett. Für Physiker: Dekohärenzprinzip. Ist der Mond noch da, wenn man wegschaut? Aber ja doch, weil der Rest des Universums immer noch hinschaut! Wenn Sie es schaffen den Mond vom ganzen Rest des Universums abzukoppeln können sie ihn durch einen Doppelspalt wabern lassen. Schaut niemand hin passieren sofort seltsame Dinge, also brauchen wir ein grundlegendes Beziehungsgefüge um von Wirklichkeit reden zu können. Grundlegend wäre die Beziehung der Identität und Nichtidentität: uniREL := [(=); (#);];
Dies nennen wir Universalbeziehung in unserem Universum oder auch das Fundamentalgefüge. Ein anderes Gefüge bildet die Beziehung zwischen Element und Noptronischer Menge hier definiert als reines Objekt:

object=$.X: [X*p :< $.X];
object=X*p: [X*k :< X*p];

Mehr Abstraktion bitte!
(Blöder Abstraktions-Heini! Ist doch so schon alles schlimm genug! Anm. der Redaktion)

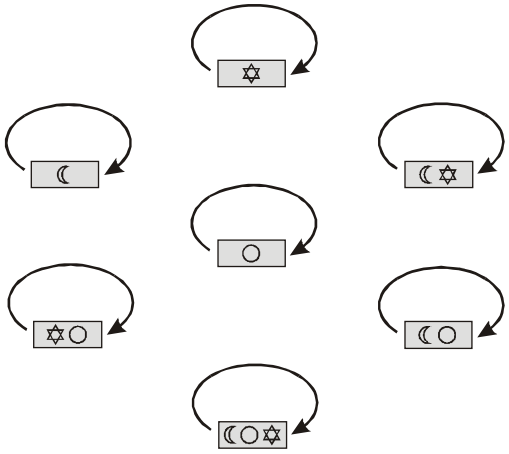
Wie wollen wir unsere Objekte benennen? Es handelt sich wohl doch um noptronische Quanten. Das heißt: Die kleinsten und diskreten Teilchen, also Atome, eines übergeordneten Systems, definieren sich noptronisch, also über das Beziehungsgefüge des zu definierenden Teilchens. Das System selbst besteht aus dem gesamten Beziehungsgefüge aller Quanten und wird damit wieder zu einem eigenständigen Objekt welches wiederum in einem anderen Beziehungsgefüge eingebettet sein kann.
Auf jeden Fall fordern wir die "Identität der Objekte": Jedes Objekt ist mit sich selbst identisch und was mit sich selbst identisch ist sei ebenfalls ein Objekt und muß damit als Beziehungsgefüge von Objekten beschreibbar sein.
Und wir fordern die Unterschiedlichkeit von Objekten: Unterschiedliche Objekte unterscheiden sich von einander. Ob zwei Objekte gleich oder nicht gleich sind läßt sich nur an ihrer Definition erkennen, daher sei ein Objekt nur dann ein existierendes Objekt, wenn wir seine Definition angeben können. Andernfalls ist es ein hypothetisches oder protothetisches Objekt.

Die 7 Kostbarkeiten seien all jene Objekte, welche mit der Universalbeziehung [(=); (#);] und einem angenommenen Unding RA, welches jedoch kein Objekt sei, erzeugt werden können. Sie sollen als Startwert einer noptronischen Definition von Cocos*Palam dienen:

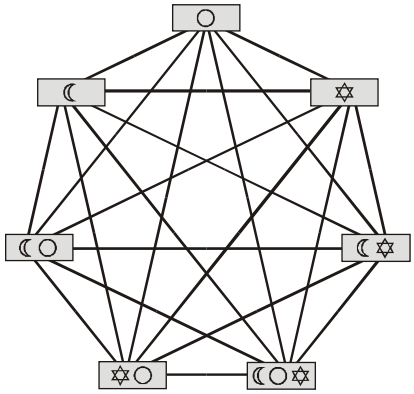
○ := [RA # RA];
✡ := [x = x];
☾ := [x # y];
✡○ := [x = x; RA # RA;];
☾○ := [x # y; RA # RA;];
☾✡ := [x # y; x = x;];
☾○✡ := [x # y; RA # RA; x = x;];

RAL := [RA # RA];
Myn := [x = x];
Mon := [x # y];
Mayn := [x = x; RA # RA;];
Moon := [x # y; RA # RA;];
Palam := [x # y; x = x;];
Cocos := [x # y; RA # RA; x = x;];

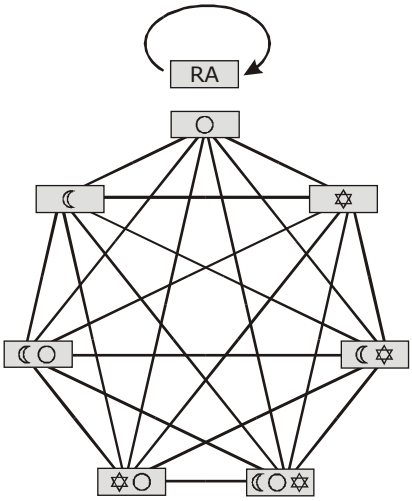
Myn:



Mon:



Moon:



RAL:

RAL := [RA:Mon:RA];
Myn := [x:Myn:x];
Mon := [x:Mon:y];
Mayn := [x:Myn:x; RA:Mon:RA;];
Moon := [x:Mon:y; RA:Mon:RA;];
Palam := [x:Mon:y; x:Myn:x;];
Cocos := [x:Mon:y; RA:Mon:RA; x:Myn:x;];

Noptronisch codiert nach einem Orakel von Dr. Rosenquarz:

```
matrixSET=Cocos*Palam:
{Myn, Mon, Mayn, Moon, Cocos, Palam, RAL};
RAL := { };
Myn := X*p;
Mon := X'*p;
Mayn := $.X;
Moon := $.X';
Cocos := {Myn, Mon, Mayn, Moon, Cocos, RAL};
Palam := {Myn, Mon, Mayn, Moon, Cocos};
```

Wobei hier ein zu X komplementärer Sternwander X' in NOPTONIC global definiert sei:

```
DEF=X':
[
var: p, q;
X' := [X'*p := $.X <- X*q:type=[X*q :< X*p];
];
```

(HaHa, da hat jemand den mengentheoretischen Kontrapunkt entdeckt! Anm. der Redaktion)

Achtung! Hier findet eine Matrixtransformation statt, denn beim Schreiben des starBUG werden alle Elemente aus $.X <- X*q:type=[X*q :< X*p] automatisch in den neuen Kontext X' überführt. Außerhalb des starBUG muß dies ebenso wie beim benannten Sternwander artikuliert werden.

Als reflexive Monade codiert:

```
$.CX-Palam := Cocos*Palam = {Myn, Mon, Mayn, Moon, Cocos, Palam, RAL};
X-Palam*ALPHA := RAL = {Palam};
Myn := X*p <+ RAL;
Mon := X'*p <+ RAL;
Mayn := $.X <+ RAL;
Moon := $.X' <+ RAL;
Cocos := {Myn, Mon, Mayn, Moon, Cocos, RAL};
X-Palam*OMEGA := Palam = {Myn, Mon, Mayn, Moon, Cocos};
```

Damit haben wir Cocos*Palam mengentheoretisch in seiner grundlegenden Form modelliert! Das stellt eine intellektuelle Leistung dar, die sich durchaus mit dem Fisch-Schwanz-Genuckel von Paris Hilton messen kann.
Wer war Paris Hilton? Paul fragt warum der tote Bär gar kein Fell hat. Wer ist eigentlich Paul? Ich finde Russ Meyer sollte den Papst heilig sprechen. Egal welchen. Und dem Einstein spendier' ich 'ne Runde im Kinderpuff, alle Kinder sind natürlich durch schnelles hin & her Fahren über 1000 Jahre alt. Alles völlig legal. Sehen aber aus wie..... hö hö, wollen wir jetzt hier nicht zeigen sonst kommen die Pädophilenhasser und ficken mich zur Strafe in den Popo bis es juckt und blutet. Amen und Amen und Amen.
Nun können wir aber Konstrukte wie beispielsweise "Großer Sternenkreis" und andere Objekte in Cocos*Palam mengentheoretisch ausdrücken: [Tacan:Cirin]:type=mayor := $.X <+> $.X';
Oder noch besser: Wir sind jetzt in der Lage neue Objekte durch mengentheoretische Operationen wie POPs, Bildung einer noptronischen Teilmenge usw überhaupt erst zu erzeugen um eine neue, differenzierte Einsicht in die Feinstruktur von Cocos*Palam zu gewinnen.
Grundlage dazu sei unsere Monade $.CX-Palam mit ihrem Sternwanderkern $.X der mit seiner Gradation sogar schon eine Art primitiver Zeit ermöglicht.
Auch Symbolketten (symbolSTRING=(): <stringBLUBBER>;) erhalten nun einen realen Sinn indem sie noptonisch codiert werden können: symbolSTRING=☾○✡: <Cocos>;
Mit diesem Rüstzeug wird symbolisches Gejockel aus 1001 Nacht zu einem mathemagischen Gedankengebäude aus allerhöchster logischer Präzision und rein magisch mystischer Inspiration.
Cocos*Palam := [Nichts bleibt schön als das Erfundene, alles andere stiehlt die Zeit....];

Wir können nun mit Hilfe der Theorie der Objekte eine Menge mit Noptronischer Konsistenz über die Beziehung (:<) oder ("ist Element von") beschreiben. Wenn jemand die Frage stellt, was denn eine Menge sei, so können wir antworten:

```
DEF=set:
[
object :< set; => object:type=set;
];
```

Achtung! Hier wird vom Objekt die Eigenschaft von set gefordert, was bedeutet, daß alle Elemente einer Menge wiederum Mengen sind. type=set bewirkt, daß set hier lediglich die Mengeneigenschaft vom Objekt darstellt und das Objekt selbst nicht die zu definierende Menge sein muß, aber sein kann, was uns dann zu einer primär reflexiven Menge führt.

Die Menge $.X besteht nun aus folgenden Beziehungen: set=$.X: [[X*q :< X*p] :< $.X;];

Da jedes Element von $.X ebenfalls nur Elemente aus $.X enthält (Noptronische Konsistenz) können wir tatsächlich diese Menge durch ein Beziehungsgefüge definieren, welches gänzlich in sich selber ruht und nicht auf Informationen oder Beziehungen außerhalb seiner Selbst angewiesen ist. Die Beziehung (:<) ließe sich auch mit (<>) abstrahieren. Beispiel sei $3.X-Matrix:

```
set=$3.X-Matrix:
[
[X*2 :< X*1; X*3 :< X*1;] :< $3.X;
[X*1 :< X*2; X*3 :< X*2;] :< $3.X;
[X*1 :< X*3; X*2 :< X*3;] :< $3.X;
];
```

Aber wie könnte man die leere Menge die ja keine ihr innewohnende Objekte besitzt als Gefüge von Relationsoperatoren definieren? Die Leere Menge ist eine ganz besondere Menge und muß auch gesondert definiert werden:

```
DEF=zset:
[
set=zset: [object :< zset; => object=RA;]
];
```

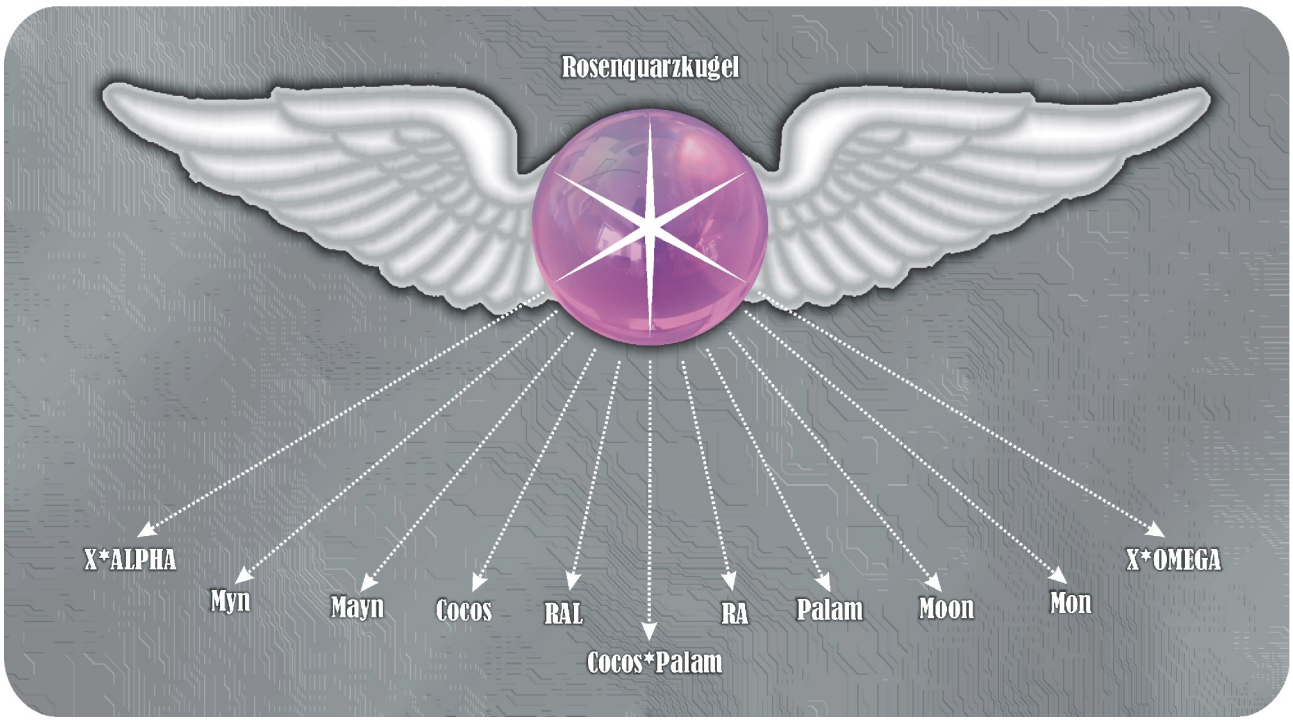
Wobei hier durch die Wertezuweisung object=RA; das Objekt als solches verschwindet, da RA kein Objekt ist und nur in einem Beziehungsgefüge zu sich selbst vorkommen darf. Hier handelt es sich jedoch um eine Wertezuweisung die eine Annihilation des Objektes erwirkt. So erhalten wir sauber und exakt die leere Menge. Es gilt natürlich: zset = colX*p:type=[X*p#X*p] = { };

Abb. rechts:
Man kann sehr gut die Kugelform in der original ägyptischen Reliefdarstellung erkennen.
Diese alten Ägypter wußten damals schon, daß die Sonne eine Kugel und keine Scheibe ist!
Allerdings war ihr Abstraktionsvermögen weit genug entwickelt, um die reine Kugel als Ideal einer universalen Ganzheit zu verehren.
Es zeichnet sich hier klar eine prototypische Realisierung des Kultes der Rosenquarzkugel ab und hätten die damals auch noch gewußt, daß die Rosenquarzkugel einen Stern hat, wären die Pyramiden sogar ohne Außerirdische noch viel viel größer geworden!

Prinzessin Chich neben der dicken Rosenquarzkugel des Pharao. Foto with courtesy by Echnaton.

```
DEF=omset:
[
set=omset: [object :< omset; => object=omset;]
];
```

Was nach dem 2. Axiom von Dr. Finsler bedeutet, es existiert außerhalb einer Sternwandermenge nur genau eine Menge omset. Ebenso existiert nur genau eine Leere Menge, obwohl RA kein Objekt ist, aber das Beziehungsgefüge der Leeren Menge wohl ein Objekt darstellt, weil wir sie als eine Menge definiert haben und sie daher mit sich selbst identisch sein muß und nicht von sich selber verschieden sein kann. Oder anders ausgedrückt, obgleich formal nicht beweisbar ist, daß nur genau eine Leere Menge existiert, so ist dies, da wir sie als ein Objekt auffassen, doch evident. Vergleichsgrundlage von Objekten sei lediglich ihr reines Beziehungsgefüge!
Es stellt sich grundsätzlich die Frage wie die Unterschiedlichkeit oder Gleichheit von Objekten zu entscheiden ist. Hier würde sich eine Art Extensionalitätsaxiom ähnlich wie in der ZF-Mengenlehre anbieten. Ist hier etwa Gödel's Knödel am Werk? Bleibt einem nichts anderes übrig, als daran zu glauben, nur weil es einfach so ist? Ja! Die Leere Menge ist der Knuff Gottes in die Mathematik! Das ist wie bei Adrenochrom, es wirkt nur dann halluzinogen, wenn man es dem Probanten vor der Verabreichung mitteilt. Trotzdem ist es kein Placeboeffekt, der ist nämlich sehr viel schwächer ausgeprägt und hat kaum eine psychedelische Wirkung. Sex, Drogen und Mengen sind eine sehr schöne Sache. Darauf nun ein "Black Elephant" von Carlsberg! Amen und Amen und Amen.



Echnaton wäre bestimmt hellauf begeistert von diesem präzisen Holismus mit cyanotisch marzipanigem, mystisch-magischen Beigeschmack! So genaue Konstruktionsanleitungen gibt's nicht alle Tage, höchstens alle 6000 Jahre und auch nur, wenn die Götter huldig gestimmt sind, sagt der Pink Panther: "Heute ist nicht alle Tage, ich komm' wieder, keine Frage!"

(Ein sprechtechnisches Kabinettsstückchen oder was? Was weiß der Depp schon von dem Echnaton, da hat der ja noch gar nicht gelebt. Echnaton wurde 2 Jahre nach seiner Geburt von Außerirdischen entführt und 10 Jahre auf der Vega unterrichtet. Er machte dort seinen Master in Noptronischer Theorie und kehrte dann nach Ägypten zurück. Das ist Alles. Außerdem gibt es gar keine Außerirdischen, sonst wären die schon längst gelandet und würden defekte Knusperriegel verteilen die seltsam schmecken und nach deren Genuß man geräuschlos herumfliegen kann. Anmerkung des Übersetzers)

Die Menge $.X := {X*1, X*2, X*3, X*4, ..., X*n} bezeichnen wir als Sternwandermenge X im Grade $. Der Buchstabe X steht hier für CHI, denn für frei definierte Sternwander ist der Buchstabe CHI reserviert. Fest definierte Sternwander werden mit großen griechischen Buchstaben benannt. Die Menge $.X sei unsere Matrixmenge, die Elemente von $.X heißen Definitionsmengen oder auch Sterne der Matrixmenge $.X
Damit auch wirklich Guthe Sternwanderey einsetzt, muß der Sternwander folgenden geforderten Grundsätzen genügen:

(1) $.X := {X*1, X*2, X*3, ..., X*n}
(2) [r > q] <=> [|$r.X| > |$q.X|]
(3) $.X*p < $.X
(4) |$.X| := CARD*$
(5) $0.X := { }

oder als globale Definition in NOPTONIC:

```
DEF=GUS:
        [
        var: p, q, r, Axiom;
        NOP=GUS: 5.[OP=index.(): Axiom] =
                [
                OP=1: $.X := colX*p;
                OP=2: [r > q] <=> [|$r.X| > |$q.X|];
                OP=3: $.X*p < $.X;
                OP=4: |$.X| := CARD*$;
                OP=5: $0.X := { };
                ];
        ];
```

Hinweise zu den Forderungen der Guthen Sternwanderey:
Begründung von (1):
Das Matrixmengenpostulat behauptet die Existenz einer Menge $.X mit allen Konsequenzen, die eine Menge mit sich bringt: Alle Elemente der Menge sollen voneinander wohl unterschiedene (Unterscheidungskalkül) und wohl definierte Objekte (Existenzkalkül) darstellen, ebenso soll $.X dem 1. Axiom der Mengenlehre von Dr. Finsler genügen und damit nach der Cantor'schen Mengendefinition eine echte Zusammenfassung sein.
All diese Eigenschaften von $.X dürfen wir bei der Matrixmengensynthese als Voraussetzung verwenden und falls sich kein Widerspruch herleiten läßt, gilt die Wohlunterschiedenheit und damit auch die Wohldefiniertheit der einzelnen Objekte als gesichert. Da dies nun nicht selbstverständlich ist, erscheint mir diese Forderung keineswegs trivial oder tautologisch angemuggert. Bekanntestes Beispiel für eine Trivialität ist das legendäre Radio Eriwan, das immer Recht hat, was ja schließlich Radio Eriwan auch verkündet und damit sich selbst bestätigt (siehe auch Curry-Paradoxon). Nein, hier besteht im Gegensatz zu Radio Eriwan die Möglichkeit, daß die Matrixmengensynthese nicht gelingt weil definierte Elemente trotz der Annahme ihrer Verschiedenheit miteinander identisch sind (ERROR=000E: Matrix_Set_Quantum_Loss;) und damit entsteht eine Art Hintergrund, welcher der gelungenen Matrixmengensynthese eine echte Wirklichkeit verleiht. Es handelt sich also hier nicht um einen Zirkelschlußfehler, auch wenn das zunächst so aussieht. Die hypothetisch angenommene Matrixmenge wird so zu einem tatsächlichen, mathematischen Objekt. Ohne die Annahme (1) fallen imaginäre (vollständig unfundierte) noptronische Mengen komplett zu omset := {omset}; zusammen (Omega-Kollaps). Die Raumzeit und jede Form von Materie verschwindet dann in dieser Singularität. Schlimmer wie schwarze Löcher, die werden ja immer größer, je mehr man reinsteckt. Eine Omega-Singularität wird nur noch von der leeren Menge übertroffen. Da verschwindet sogar die Singularität in der Vollkommenen Leere.

Hinweis für Platoniker: Die Matrixmengensynthese ist im Grunde genommen kein Herstellungsprozeß einer Menge, denn allein mit der algorithmischen Definition des Sternwanders existiert diese Menge unmittelbar, wir erkennen sie nur noch nicht. Die Matrixmengensynthese wird hier zu einem Erkenntnisprozeß und führt uns bei Gelingen zum Wissen, daß diese Menge existiert, was durch die Kenntnis ihrer präzisen Struktur unbestreitbar belegt sei. Wird eine Noptronische Menge durch einen Benutzer (X := [user];) während des Starbugs erzeugt, könnte man eher von einer Mengensynthese sprechen. Hier fallen Erkenntnisprozeß und Synthese aufgrund der Auswahl des Benutzers der Definitionselemente aus der hypothetischen Matrixmenge zusammen.

Begründung von (2):
Eine größere Gradation erfordert auch eine größere Mächtigkeit der neuen Sternwandermenge. Wir erwarten Fortschritt! Jede noptronische Menge kann durch Quantenpermutationen in eine andere, gleichmächtige umgewandelt werden. (Beweis!)
Ein Beispiel dafür ist die erste unendliche Ordinalzahl, welche die Menge aller endlicher Ordinalzahlen ist, die man durch eine algorithmisch beschreibbare Quantenpermutation in ihren Nachfolger umwandeln kann, da beide gleichmächtig aber natürlich nicht identisch sind.
Unterschiedliche Gradationen von Sternwandermengen sollen grundsätzlich auch unterschiedlich mächtige noptronische Mengen erzeugen.

Begründung von (3):
Forderung nach der Noptronischen Konsistenz. Alle Elemente einer noptronischen Menge sind Teilmenge dieser Menge. Das holistische Prinzip wird hier radikal mengentheoretisch verwirklicht. Hermeneutischer geht's nicht.
Hier lohnt ein wenig innezuhalten um mal nachzudenken. Wäre es nicht ultraphantastisch den Mengenbegriff grundsätzlich auf Noptronische Mengen radikal zu reduzieren? Menge ist nur was einer definierbaren Matrixmenge angehört und die Matrixmenge selbst! Einfach ein paar Dinge zusammenfassen macht noch keine Menge im noptronischen Sinn. Wir fordern ein strukturelles Beziehungsgefüge, ein Ding das sich selber zusammen hält aufgrund seiner geistigen Kohärenz. Was impliziert das Matrixmengenpostulat? Es setzt uns eine ungeformte Ur-Masse vor (Ylem, Chaos, RAL) die wir durch Auswahl der Elemente formen. Entweder wir als user oder ein Algorithmus arbeitet wie ein Bildhauer aus einer nicht nur ungeformten sondern auch aus einer lediglich hypothetisch angenommenen Masse eine Skulptur heraus. Das klingt zwar zunächst wie "Creatio ex Nihilo", aber wir erschaffen ja durch die Eingabe einer Information ein Etwas. Der Gag an der ganzen Sache ist der, daß die Information selbst die Matrix ist in der die Information codiert wird. Ein Datenspeicher der gewissermaßen selbst aus den in ihm gespeicherten Daten besteht!

Begründung von (4):
Es soll eine Funktion existieren, die eine Beziehung zwischen der Gradation und der Mächtigkeit der zu erzeugenden Sternwandermenge beschreibt. Um einen Noponator auszuführen braucht der Magische Zentralcomputer vorher die Anzahl der anstehenden Operatorindizes.

Begründung von (5):
Der global definierte Noponator der den Starbug ausführt und damit den Definitionscluster erzeugt wird später definiert. Wenn wir diesen Noponator annehmen ergibt sich für $=0 :

starBUG=0.X:
0.[OP=index.X: X*p := X*p:norm;] = [OP=(): ERROR="no_operation_performed";] = [];

Für OP=index.X gibt der Magische Zentralcomputer gleich den Leeroperator OP=() aus, da die Indexvariable nicht den Wert 0 annehmen kann. Starbug ist als NOP angelegt, also wird obwohl der Noponator nicht ausgeführt wurde, ein NOP angelegt in dem keine Operation in Bezug auf X stattfindet. Der Leeroperator OP=() macht es dem Magischen Zentral Computer möglich dies zu artikulieren und die entsprechende Meldung auszugeben.
Es wird daher kein Glied definiert (oder der mysteriöse RA-Genius X*0).
[] ist das sogenannte Leer-NOP oder auch "die Idee des NOPs" und kann ohne Probleme auch mit der leeren Menge { } identifiziert werden. Aber was ist der Unterschied zwischen NOP und Menge? Wenn wir die Menge wirklich als rein noptronisches Konstrukt auffassen, sind NOPs eher das, was man mit einem Komprehensionsprinzip leisten kann. Oder einfach mal was zusammensammeln. Aber die Gritli meint, der noptronische Mengenbegriff wäre viel knackiger und auch universeller.

Man könnte bei der leeren Menge auch sagen, egal welcher Algorithmus nicht angewendet wird, das Ergebnis ist immer, daß keine Operation stattfindet was mit dem Leeroperator OP=() nur beschrieben wird.
So stellt die leere Menge oder auch $0.X eine Art Sternwander-Singularität dar. Wobei in dem Ausdruck $0.X die Information enthalten ist welcher Algorithmus nicht angewendet werden soll (nämlich X), während die leere Menge uns keine Auskunft darüber erteilt aus welchem Algorithmus, der nicht angewendet wird, ihre Inhaltslosigkeit resultiert!
Eine weitere Frage die sich bei (5) stellt ist die nach der Noptronischen Konsistenz von $0.X Ist die leere Menge eine noptronische Menge? Dazu müßte (3) für all ihre Elemente gelten, aber die leere Menge hat ja gar keine Elemente! Wenn wir einfach annehmen, daß die leere Menge eine noptronische Menge ist würde das zumindest dem Potenzmengentheorem nicht widersprechen, denn die Potenzmenge der leeren Menge ist sicher noptronisch.
Interessant wäre hier, ob aus der noptronischen Konsistenz einer Potenzmenge auch die noptronische Konsistenz ihrer Basismenge folgt (Beweis!). Dann könnte man ja die Annahme, daß $0.X noptronisch ist herleiten. Sehr praktisch. Und das Potenzmengenaxiom in ZF wäre auch überflüssig. Je weniger Axiome, desto besser das System. Je kleiner und schäbiger das Hotelzimmer, desto besser der Fick. Amen & Halleluja.

Nun beginnen wir gleich mit einem einfachen aber höchst bemerkenswerten Sternwander dessen generierte Sternwandermengen Mathematiker unter dem Namen "von Neumann-Zahlen" sicher gut kennen. Es sei ein freier und unbenannter Sternwander X definiert durch:

X := [X*(p+1) := X*p <+ X*p; X*1 := { }; |$.X| := $;];

Damit ist die Sternwandermenge im Grad $ laut Axiom (1) zumindest hypothetisch festgelegt:

$.X := colX*p = {X*1, X*2, ..., X*n};

Wir erzeugen sukzessiv mit dem Sternwander X bei X*1 := { }; beginnend die Definitionsmengen. Die Leere Menge ist als Startelement gut geeignet, da sie nach Axiom (5) selbst eine Sternwandermenge ist. Wir brauchen also keine neuen Objekte einzuführen.
Mit X sollten alle Elemente von X*1 bis X*n der Sternwandermenge $.X festgelegt sein, die Kardinalität der Sternwandermenge im Grad $ ist aber auf jeden Fall jetzt schon bekannt. Somit ist (4) erfüllt.
Der Magische Zentralcomputer berechnet zunächst den Definitionscluster, ein NOP in dem alle Elemente von $.X in der übersichtlichen Orthonormalform generiert werden und damit auch exakt definiert sind. Hier die in NOPTONIC allgemeingültige Definition des Noponators starBUG:

```
DEF=starBUG:
[
var: n, p, q;
p=index; n=|$.X|; q=$;
starBUG=q.X: n.[OP=index.X: X*p := X*p:type];
];
```

Setzen wir einmal $=9; und lassen den starBUG 9.X in der Orthonormalform berechnen:

```
starBUG=9.X:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := { };
OP=2.X: X*2 := {X*1};
OP=3.X: X*3 := {X*1, X*2};
OP=4.X: X*4 := {X*1, X*2, X*3};
OP=5.X: X*5 := {X*1, X*2, X*3, X*4};
OP=6.X: X*6 := {X*1, X*2, X*3, X*4, X*5};
OP=7.X: X*7 := {X*1, X*2, X*3, X*4, X*5, X*6};
OP=8.X: X*8 := {X*1, X*2, X*3, X*4, X*5, X*6, X*7};
OP=9.X: X*9 := {X*1, X*2, X*3, X*4, X*5, X*6, X*7, X*8};
];
```

Die Sternwandermenge $9.X läßt sich tatsächlich in Orthonormalform darstellen, was bedeutet, jedes beliebige Element X*p aus $9.X ist selbst eine Menge, welche nur Elemente aus $9.X enthält oder anders ausgedrückt, jedes Element X*p aus $9.X ist eine Teilmenge von $9.X Auf jeden Fall ist damit Axiom (3) oder die Forderung nach der Noptronischen Konsistenz für Sternwandermengen erfüllt. Gilt das aber auch für alle Werte von $? Dazu kommen wir später. Ohne Fehlermeldung des Magischen Zentralcomputers sind alle Elemente aus $9.X nun nicht mehr nur geforderte oder hypothetische Elemente, sondern tatsächliche, wohldefinierte, mathematisch existierende Objekte.
Dieser Vorgang heißt Matrixmengensynthese und ist die Garantie $9.X als existierende Menge bezeichnen zu können. Alle Forderung der Guthen Sternwanderey werden offenbar erfüllt und wir dürfen schreiben: $9.X = {X*1, X*2, X*3, X*4, X*5, X*6, X*7, X*8, X*9};

Es sei noch erwähnt, daß dieser Definitionscluster, also das hier erzeugte NOP, nicht identisch ist mit der Menge $9.X, denn der vollständige Definitionscluster, also die erfolgreiche und ohne Fehlermeldungen ausgeführte Matrixmengensynthese bedeutet, daß diese durch unseren gewählten Starbug-Noponator grundsätzlich ausführbar ist und damit ein NOP erzeugt wird. Für die Fans von "Algorithmen bei der Arbeit" sei erwähnt, daß die Rechenschritte zur Erzeugung der einzelnen Elemente nicht im Definitionscluster auftauchen müssen, dies sei nicht die Aufgabe des Definitionsclusters, sondern die Darstellung der Definition eines Elementes X*p aufgrund eines tatsächlich existierenden und dem Element zugeordneten Operators OP=p.X, der das Vorhandensein unserer Rechenschritte und damit die operative Abgeschlossenheit die zu X*p führen repräsentiert.
Unser Magischer Zentral Computer hat genug virtuelle Schmierzettel (RAM) um die Elemente nach den Anweisungen des vorgegebenen Sternwanders X in einer praktikablen Programmiersprache zu erzeugen. Natürlich könnte der Leser auch selbst mit Bleistift und Papier bewaffnet die Elemente berechnen und in den Definitionscluster in der geforderten Form eintragen. Wie steht es mit Axiom (1) wenn wir auch daran denken, daß eine korrekte Cantor'sche Menge die Unterschiedlichkeit all ihrer Elemente fordert? Nun, wenn sich wirklich alle Elemente aus $9.X gemäß Axiom (1) voneinander unterscheiden, folgt aus unserem Definitionscluster unweigerlich, daß sich in der Tat alle Elemente aus $9.X voneinander unterscheiden!
Das erscheint unserer Vernunft wie die Stromgewinnung indem man einfach das eine Ende des Verlängerungskabels in das andere steckt und nun erwartet, daß Strom fließt und sicher weiß jeder was passiert, wenn der Vakuumsauger im Film Yellow Submarine damit beginnt sich selber aufzusaugen, weil sonst nichts mehr da ist.... Doch keine Panik, es gibt einen Ausweg. Erst einmal bezeichnen wir $9.X als “Reale Sternwandermenge”, da sich jedes ihrer Elemente auch ohne Verwendung eines beliebigen Mengenterms der Elemente aus $9.X enthält darstellen läßt, was wir durch manuelle Umformung des in orthonormalform vorliegenden Definitionsclusters leicht nachvollziehen könnten. Oder etwas bequemer und schneller, indem wir den Magischen Zentralcomputer einfach durch eine Typenangabe die Realform der Elemente berechnen lassen:

```
starBUG=9.X:
9.[OP=index.X: X*p := X*p:real] =
[
OP=1.X: X*1 := { };
OP=2.X: X*2 := {{ }};
OP=3.X: X*3 := {{ }, {{ }}};
OP=4.X: X*4 := {{ }, {{ }}, {{ }, {{ }}}};
OP=5.X: X*5 := {{ }, {{ }}, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {{ }}}}};
OP=6.X: X*6 := [ERROR="invalide_wrtln"] = X*6;
OP=7.X: X*7 := [ERROR="invalide_wrtln"] = X*7;
OP=8.X: X*8 := [ERROR="invalide_wrtln"] = X*8;
OP=9.X: X*9 := [ERROR="invalide_wrtln"] = X*9;
];
```

Die letzten 4 Elemente werden wegen ihrer enormen Länge nicht ausgedruckt, was auch gar nicht notwendig ist. Aufgrund des Sternwanders ist die Verschiedenheit aller Elemente aus $9.X wenn sie in Realform vorliegen evident, daher verzichten wir hier auf einen formalen Beweis. Sogar die Erfüllung von Forderung (2) läßt sich hier leicht einsehen, der Beweis sei dem Leser überlassen.
Man sieht aber hier schon wie sich mit einfachsten Sternwanderoperatoren unübersichtliche und stark verschachtelte Mengen bilden können. Spätestens bei Element $9.X*5 verliert man bei der üblichen Mengenschreibweise den Überblick und es treten Symptome wie Augenflimmern, Bildzittern oder Gefühle von Ohnmacht und Hilflosigkeit auf. Die grafische Darstellung eröffnet glücklicherweise etwas mehr Anschaulichkeit der hier geschaffenen Noptronischen Menge. Einmal können wir mit einem Pfeilediagramm, wie es auch in der Graphentheorie verwendet wird, uns etwas Überblick von diesem verschachtelten Mengenhokuspokus verschaffen. Der Pfeil übernimmt hier die Funktion “ist Element von”: [X*p ---> X*q] := X*p :< X*q;
Eine weitere mögliche Darstellung, gut geeignet um Strukturen von sehr großen Werten für n optisch darzustellen ist die Binärmatrix. Dabei handelt es sich um ein einfaches Pixelquadrat, wobei die Zeilen die Definitionsmengen repräsentieren und jedem Pixel den Wert 1 oder 0, b.z.w. schwarz oder weiß zugeordnet wird, gemäß der Funktion “ist Element von”.

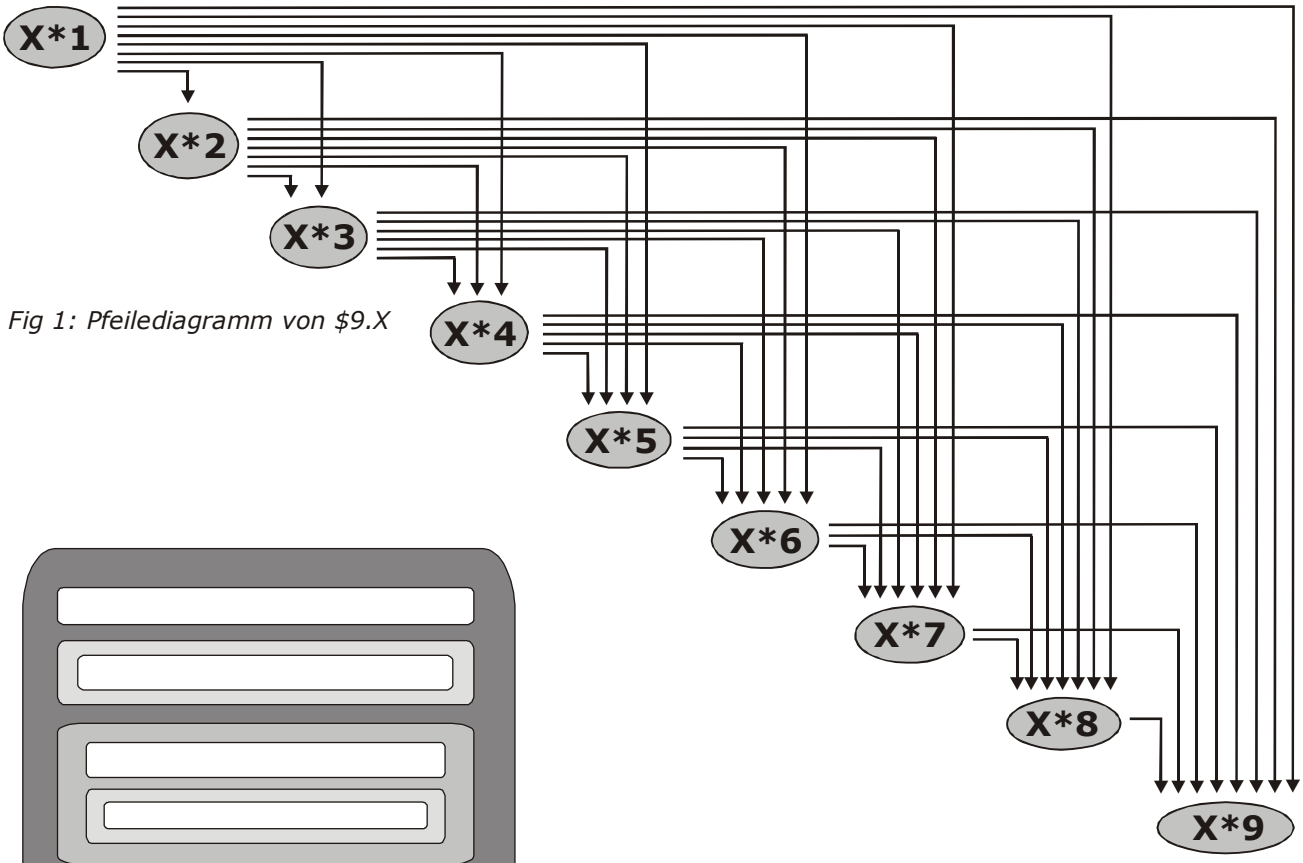


Fig 1: Pfeilediagramm von $9.X

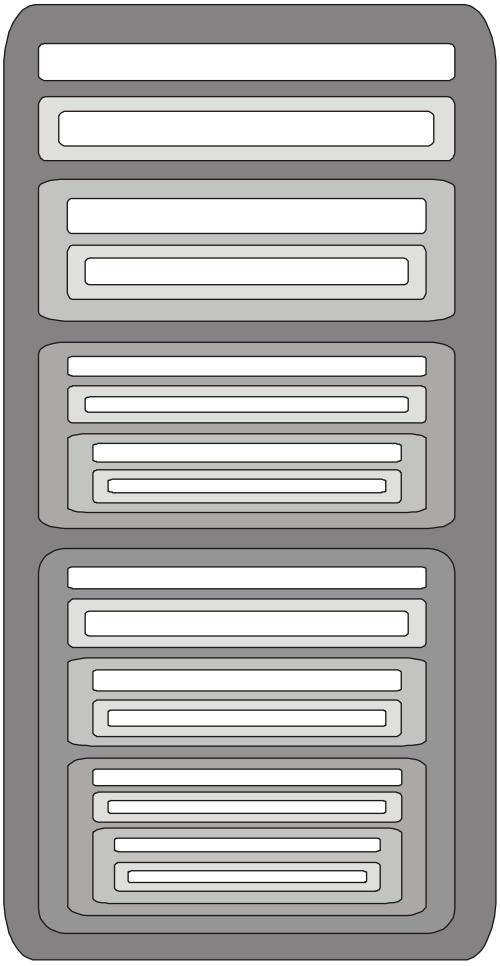


Fig. 3 : Schachtelgraph von $5.X

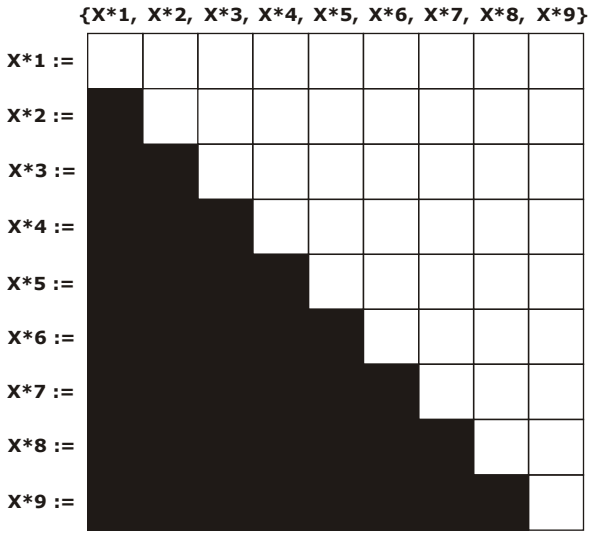


Fig. 2 : Binärmatrix von $9.X

Fig. 3 zeigt eine andere Darstellung, die im wesentlichen eher die ästhetischen Aspekte solcher Mengen, hier für $=5; veranschaulicht. Die Abstufung von Grau und weiß dient der Übersicht und betont die Ästhetik der Selbstähnlichkeit, denn man erkennt schnell diese Tendenz, was aufgrund der Sternwanderfunktionsvorschrift nicht überrascht.
Wäre der Wert von n unendlich, so wäre dieses Object in der Tat selbstähnlich, das hieße, ein bestimmter Ausschnitt daraus wäre mit dem gesamten Object identisch. Man denke an die reflexive Eigenschaft von unendlichen Mengen oder die Landkarte welche in der Landkarte enthalten ist. Siehe auch: *Bertrand Russell*, "Einführung in die mathematische Philiosophie".

Es sei nun wieder ein freier unbenannter Sternwander X gegeben durch:

X := [X*(p+1) := X*1 <+ X*p; X*1 := { }; |$.X| := $;];

Daraus ergibt sich die hypothetische Sternwandermenge:

$.X := {X*1, X*2, X*3, ..., X*n};

Wir setzen $=9 und lassen uns diesmal die Menge gleich in Realform ausgeben, denn es handelt sich um ein etwas übersichtlicheres noptronisches Mengenkonstrukt:

```
starBUG=9.X:
9.[OP=index.X: X*p := X*p:real] =
[
OP=1.X: X*1 := { };
OP=2.X: X*2 := {{ }};
OP=3.X: X*3 := {{{ }}};
OP=4.X: X*4 := {{{{ }}}};
OP=5.X: X*5 := {{{{{ }}}}};
OP=6.X: X*6 := {{{{{{ }}}}}};
OP=7.X: X*7 := {{{{{{{ }}}}}}};
OP=8.X: X*8 := {{{{{{{{ }}}}}}}};
OP=9.X: X*9 := {{{{{{{{{ }}}}}}}}};
];
```

$9.X:real = {{ }, {{ }}, {{{ }}}, {{{{ }}}}, {{{{{ }}}}}, {{{{{{ }}}}}}, {{{{{{{ }}}}}}}, {{{{{{{{ }}}}}}}}, {{{{{{{{{ }}}}}}}}};

Auch hier erhalten wir eine reale Noptronische Menge, jetzt wie eine Kette, daher heißt so eine Sternwandermenge auch “String”. Es war auch sehr schön induktiv definiert. Das Ganze hat optisch gesehen etwas von einfacher Kausalität. Diese Art der Induktion könnte man daher als “Lineare Induktion” bezeichnen und entspricht, wie alle realen Sternwandermengen, der Unterlichtgeschwindigkeit und kausalen Ausbreitungsvorgängen.

Fig. 3 : Pfeilediagramm

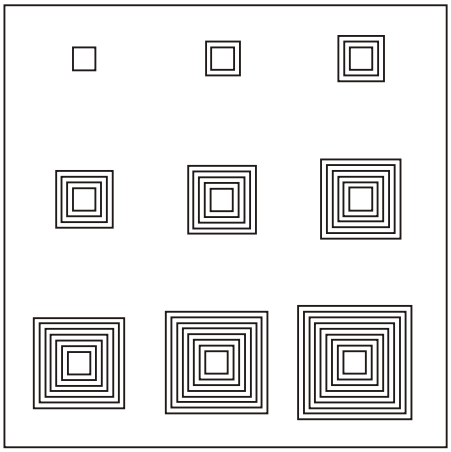


Fig. 4 : Mengendiagramm

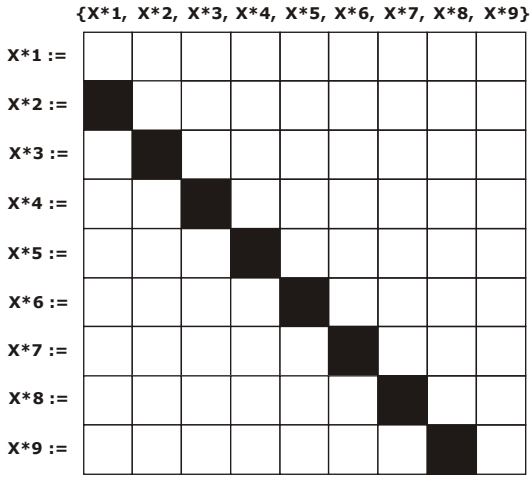


Fig. 5 : Binärmatrix

Es geht aber auch anders. Dieser Sternwander kann nicht mehr linear induktiv erzeugt werden. Hier werden die einzelnen Sterne nicht aus einem ersten Element oder aus einem “Vorgänger” entwickelt, sondern sind gewissermaßen gleichzeitig mit einem Knall da. Ursache und Wirkung verschmelzen sozusagen. Diese Art der Induktion heißt “Superinduktion” und dies entspricht genau der Lichtgeschwindigkeit.
Die daraus resultierende Sternwandermenge ist daher in Realform nicht darstellbar, denn es handelt sich um eine imaginäre Sternwandermenge, da kein einziger Stern ein reales Element enthält, egal welche Transkription und wie oft sie auch angewendet wird.
Diesen Sternwander nennen wir “Ur-Sternwander”, denn der daraus resultierende Noptronische Raum ist von einer außergewöhnlichen Symmetrie und soll später noch im Urknall gerne und viel Verwendung finden. Der Sternwander sei:

```
X := [X*p := $.X <- X*p; |$.X| := $;];
```

Setzen wir einmal $=9 und lassen den starBUG berechnen:

```
starBUG=9.X:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4, X*5, X*6, X*7, X*8, X*9};
OP=2.X: X*2 := {X*1, X*3, X*4, X*5, X*6, X*7, X*8, X*9};
OP=3.X: X*3 := {X*1, X*2, X*4, X*5, X*6, X*7, X*8, X*9};
OP=4.X: X*4 := {X*1, X*2, X*3, X*5, X*6, X*7, X*8, X*9};
OP=5.X: X*5 := {X*1, X*2, X*3, X*4, X*6, X*7, X*8, X*9};
OP=6.X: X*6 := {X*1, X*2, X*3, X*4, X*5, X*7, X*8, X*9};
OP=7.X: X*7 := {X*1, X*2, X*3, X*4, X*5, X*6, X*8, X*9};
OP=8.X: X*8 := {X*1, X*2, X*3, X*4, X*5, X*6, X*7, X*9};
OP=9.X: X*9 := {X*1, X*2, X*3, X*4, X*5, X*6, X*7, X*8};
];
```

Also gilt nach gelungener Matrixmengensynthese:

```
$9.X = {X*1, X*2, X*3, X*4, X*5, X*6, X*7, X*8, X*9};
```

Anmerkung:
Versuchen Sie sich diese Menge einmal in Form eines "Schachtelmodells" oder als Mengendiagramm vorzustellen. Das bringt die Synapsen zum Leuchten. Da kriegen Mengentheoretiker feuchte Augen und die Typen von der Typentheorie heißkalte Panikattacken. Hier zeigt sich die Stärke des Noptronischen Mengenbegriffs, da er über Intuition und einfache Anschauung hinaus in neue abstraktere Mengenvorstellungen vordringt, die noch nie ein Mensch zuvor gesehen hat. Wenn das der Bertrand noch erlebt hätte, so ein schöner Mengenhokuspokus!

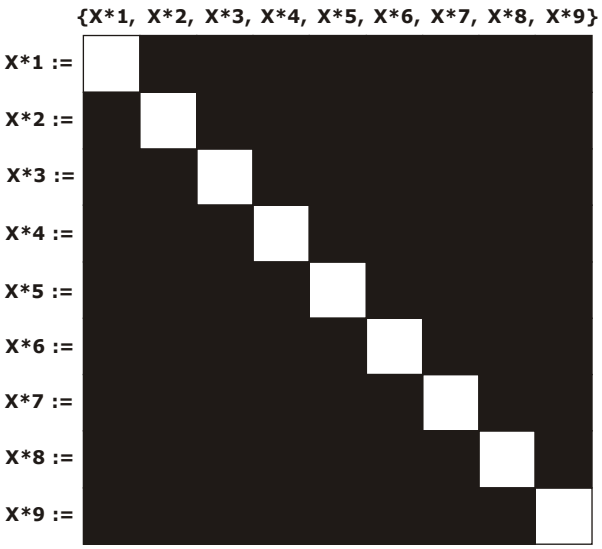


Fig. 6 : Binärmatrix der Menge $9.X

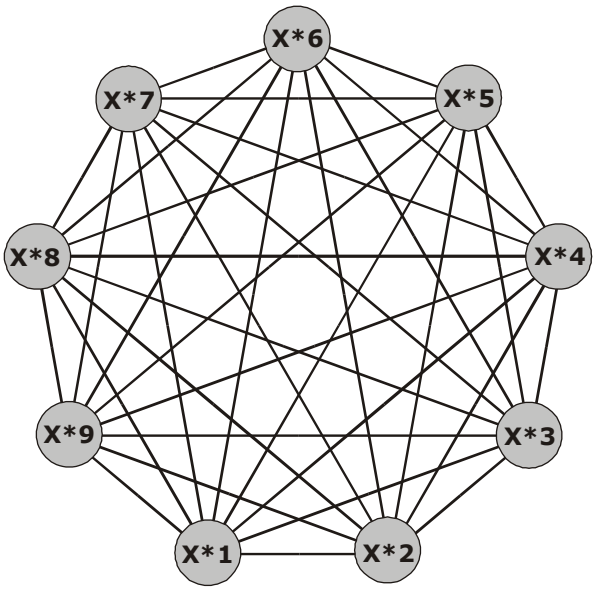


Fig. 7 : Pfeilediagramm der Menge $9.X
(Kanten ohne Spitzen sind Doppelpfeile)

Nun ein ganz anderer Sternwander, und zwar der sogenannte Potenzmengen-Sternwander.
Das Bildungsgesetz dieses Sternwanders sei:

X := [$.X := POWSET*($-1).X; $0.X := { }; |$.X| := |($-1).X|.2*1;];

Sind wir einmal sehr vorsichtig, denn eine Exponentialfunktion in iterativer Anwendung als Beschreibung der Mächtigkeit unserer entstehenden Noptronischen Menge deutet auf ein sehr rasches Anwachsen hin. Wir setzen einmal $=4 und fordern beim Magischen Zentralcomputer die Standardausgabe (Orthonormalform) an. Ist $4.X nicht noptronisch, erhalten wir eine Fehlermeldung (OP=0.X: ERROR="Mission_Impossible").
Die Funktion POWSET sei die Potenzmengenfunktion. Auf eine Menge angewendet erzeugt sie deren Potenzmenge. POWSET ist sicher irgendwo in NOPTONIC in einer schönen Programmiersprache hinterlegt. Wir müssen uns darüber jetzt keine schweren Gedanken machen.

Zunächst die Matrixmengensynthese für $=4:

```
starBUG=4.X:
16.[OP=index.X: X*p := X*p:norm] =
[
OP=01.X: X*01 := { };
OP=02.X: X*02 := {X*1};
OP=03.X: X*03 := {X*2};
OP=04.X: X*04 := {X*3};
OP=05.X: X*05 := {X*6};
OP=06.X: X*06 := {X*1, X*2};
OP=07.X: X*07 := {X*1, X*3};
OP=08.X: X*08 := {X*1, X*6};
OP=09.X: X*09 := {X*2, X*3};
OP=10.X: X*10 := {X*2, X*6};
OP=11.X: X*11 := {X*3, X*6};
OP=12.X: X*12 := {X*1, X*2, X*3};
OP=13.X: X*13 := {X*1, X*2, X*6};
OP=14.X: X*14 := {X*1, X*3, X*6};
OP=15.X: X*15 := {X*2, X*3, X*6};
OP=16.X: X*16 := {X*1, X*2, X*3, X*6};
];
```

Also gilt für unsere Sternwandermenge im Grad $4:

```
$4.X =
{
X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08,
X*09, X*10, X*11, X*12, X*13, X*14, X*15, X*16
};
```

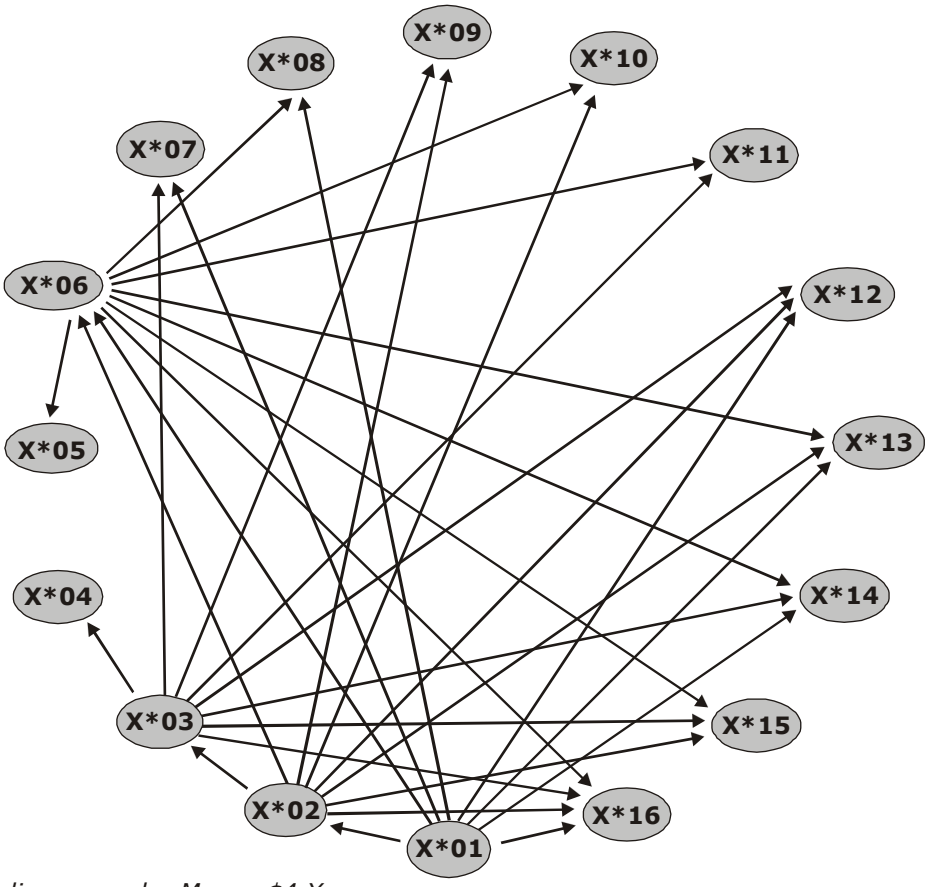


Fig. 8 : Pfeilediagramm der Menge $4.X

	{X*01,	X*02,	X*03,	X*04,	X*05,	X*06,	X*07,	X*08,	X*09,	X*10,	X*11,	X*12,	X*13,	X*14,	X*15,	X*16}
X*01 :=																
X*02 :=	■															
X*03 :=		■														
X*04 :=			■													
X*05 :=						■										
X*06 :=	■	■														
X*07 :=	■		■													
X*08 :=	■					■										
X*09 :=		■	■													
X*10 :=		■				■										
X*11 :=			■			■										
X*12 :=	■	■	■													
X*13 :=	■	■				■										
X*14 :=	■		■			■										
X*15 :=		■	■			■										
X*16 :=	■	■	■			■										

Fig. 9 : Binärmatrix der Menge $4.X

Erstaunlicherweise läßt sich $4.X in Orthonormalform darstellen. Es handelt sich hier allerdings wieder um eine reale Sternwandermenge, denn sie ist im Gegensatz zu unserem superinduktiven Sternwander auch als Definitionscluster in Realform verfügbar:

```
starBUG=4.X:
16.[OP=index.X: X*p := X*p:real] =
[
OP=01.X: X*01 := { };
OP=02.X: X*02 := {{ }};
OP=03.X: X*03 := {{{ }}};
OP=04.X: X*04 := {{{{ }}}};
OP=05.X: X*05 := {{{ }, {{ }}}};
OP=06.X: X*06 := {{ }, {{ }}};
OP=07.X: X*07 := {{ }, {{{ }}}};
OP=08.X: X*08 := {{ }, {{ }, {{ }}}};
OP=09.X: X*09 := {{{ }}, {{{ }}}};
OP=10.X: X*10 := {{{ }}, {{ }, {{ }}}};
OP=11.X: X*11 := {{{{ }}}, {{ }, {{ }}}};
OP=12.X: X*12 := {{ }, {{ }}, {{{ }}}};
OP=13.X: X*13 := {{ }, {{ }}, {{ }, {{ }}}};
OP=14.X: X*14 := {{ }, {{{ }}}, {{ }, {{ }}}};
OP=15.X: X*15 := {{{ }}, {{{ }}}, {{ }, {{ }}}};
OP=16.X: X*16 := {{ }, {{ }}, {{{ }}}, {{ }, {{ }}}};
];

$4.X:real =
{{ }, {{ }}, {{{ }}}, {{{{ }}}}, {{{ }, {{ }}}}, {{ }, {{ }}}, {{ }, {{{ }}}},
{{ }, {{ }, {{ }}}}, {{{ }}}, {{{ }}}}, {{{ }}}, {{ }, {{ }}}}, {{{{ }}}}, {{ }, {{ }}}},
{{ }, {{ }}, {{{ }}}}, {{ }, {{ }}}, {{ }, {{ }}}}, {{ }, {{{ }}}}, {{ }, {{ }}}},
{{{ }}}, {{{ }}}}, {{ }, {{ }}}}, {{ }, {{ }}}, {{{ }}}}, {{ }, {{ }}}}};
```

Man sollte an dieser Stelle erwähnen, daß die Mächtigkeit von $5.X, also der nächsten Potenzmenge, 2 hoch 16 beträgt. $5.X hat also 65536 und $6.X gar 2 hoch 65536 Elemente! Das wäre in klassischer Mengenform ausgeschrieben eine wahrhaft monströse Klammerschnur. Daß sich diese Menge, erzeugt durch Iterieren der Potenzmengenfunktion, auch in Orthonormalform schreiben läßt ist keineswegs selbstverständlich. Wenn man die Berechnung von Hand selbst durchführt, bemerkt man schnell eine Eigenart. Man muß, um zur übersichtlichen Orthonormalform zu kommen, zuerst jeden Iterationsschritt in der Real-Mengendarstellung erzeugen. Erst dann kann man diesen Term durch mühsames Vergleichen der einzelnen Glieder in die Orthonormalform überführen. Der Magische Zentral Computer kann die Orthonormalform von $4.X bei diesem Sternwander nur über die Realform errechnen.
Das Existieren einer Orthonormalform garantiert uns schließlich, daß Grundsatz (3) erfüllt ist, die eigentliche Forderung an den noptronischen Mengenbegriff, die Noptronische Konsistenz! Woher wissen wir sicher, daß für unseren hier definierten Sternwander X und für alle beliebigen Gradationen $ gilt: $.X*p < $.X ?
Aufgrund der Potenzmengendefinition ist gesichert: $.X*p < ($-1).X
Dieser Zusammenhang ist in der Tat evident, daher verzichten wir hier auf den Beweis. Denn $.X muß ja alle möglichen Teilmengen von ($-1).X enthalten, da die induktive Definition der Sternwandermenge $.X := POWSET*($-1).X genau dies für alle Gradationen $ garantiert.

Die Behauptung der obig definierte Sternwander X erzeugt für alle Gradationen aus Integer eine Noptronische Menge (Für alle Werte von $ gilt: $.X*p < $.X) ist nicht trivial und sollte daher unbedingt auch bewiesen werden.
Es bietet sich hier ein Beweis mittels Vollständiger Induktion an und ist leicht zu finden. Grundsätzlich sollte jeder Sternwander auf seine Noptronische Konsistenz hin überprüft werden. Nicht alle induktiven Verfahren liefern auch Noptronische Mengen! Grundsätzlich wäre zu bedenken welche Art von Algorithmen mit Gewißheit Noptronische Mengen generieren.

Beweis:

Empirie:

Für $1.X, $2.X, $3.X sowie $4.X ist durch empirisches Überprüfen unsere Behauptung gesichert. Sämtliche Elemente dieser vier Mengen sind Teilmengen ihrer Matrixmenge. $0.X gilt zwar laut Axiom (5) als Noptronische Menge, sei aber als Sonderfall vorsichtshalber ausgenommen.

Annahme:

Sei $.X eine Matrixmenge unseres induktiv definierten Potenzmengen-Sternwanders mit der Menge $0.X := { } als Startelement, so nehmen wir einmal an es gilt tatsächlich: $.X*p < $.X

Die Menge ($+1).X muß sich dann aus folgenden Elementen aufbauen:

1. Zunächst ist da die Leere Menge und die Menge $.X, beides Teilmengen von $.X

2. Dann müssen alle Elemente von $.X in ($+1).X auftauchen, was aus unserer Annahme und der Definition der Potenzmenge folgt:
   [$.X*p < $.X] => [$.X*p :< ($+1).X]

3. Welche Elemente müssen in ($+1).X noch vorkommen? Nach der Definition der Potenzmenge natürlich auch noch alle echten Teilmengen von $.X
   Darunter sind nach unserer Annahme alle Elemente von $.X, aber darüber hinaus möglicherweise auch Teilmengen von $.X, die in $.X als Element nicht vorkommen.
   Nennen wir diese Teilmengen ($+1).X*q, da sie schließlich Elemente von ($+1).X sind.

Conclusion 1:

a) Die Leere Menge ist sicher eine Teilmenge von ($+1).X

b) Da nach Punkt 2 alle Elemente von $.X in ($+1).X vorkommen, könnte man nun genau diese Elemente zu einer Menge M zusammenfassen und hätte damit sicher eine Teilmenge von ($+1).X erzeugt. Diese Menge M aber ist $.X, womit garantiert gilt: $.X < ($+1).X

Conclusion 2:

Alle Teilmengen von $.X die ja als Element in ($+1).X vorkommen, dürfen jedoch nach der Definition der Teilmenge nur aus den Elementen von $.X aufgebaut sein und diese müssen auch gemäß Punkt 2 *alle* in ($+1).X vorkommen, woraus unweigerlich folgt, daß jede nur mögliche Teilmenge von $.X auch eine Teilmenge von ($+1).X sein muß.
Denn alle Elemente von $.X sind nach Punkt 2 auch Elemente von ($+1).X, daher kann jedes ($+1).X*q als Teilmenge von ($+1).X dargestellt werden!

Conclusion 3:

Natürlich gilt die Überlegung von Conclusion 2 auch für sämtliche Elemente $.X*p in ($+1).X

Schlußfolgerung:

Wenn $.X von Noptronischer Konsistenz ist, dann ist es auch ($+1).X Gemäß der Empirie müssen daher alle Gradationen des Potenzmengen-Sternwanders mit dem Startwert $0.X := { } der geforderten Noptronischen Konsistenz entsprechen.
Darüber hinaus können wir nun mit Fug und Recht behaupten: Ist eine Menge noptronisch, so ist es auch ihre Potenzmenge. Und diese existiert immer (oder zumindest meistens).
Überlegung: Gelten obige Überlegungen auch für unendliche noptronische Mengen?

Hier ein Sternwander mit einer kleinen Änderung, das erste Element X*1 soll sich aus dem letzten Element X*n definieren und das letzte Element soll sich aus seinem Vorgänger ergeben.

```
X :=
[
X*1 := {X*n};
[(p+1) < n] => [X*(p+1) := X*1 <+ X*p];
[n>1] => [X*n := {X*(n-1)}];
|$.X| := $; n=|$.X|;
];
```

Wir setzen $=9 und lassen uns diesmal die Menge gleich in Orthonormalform anzeigen, denn es handelt sich um eine Imaginäre Sternwandermenge:

```
starBUG=9.X:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*9};
OP=2.X: X*2 := {X*9, X*1};
OP=3.X: X*3 := {X*9, X*2};
OP=4.X: X*4 := {X*9, X*3};
OP=5.X: X*5 := {X*9, X*4};
OP=6.X: X*6 := {X*9, X*5};
OP=7.X: X*7 := {X*9, X*6};
OP=8.X: X*8 := {X*9, X*7};
OP=9.X: X*9 := {X*8};
];
```

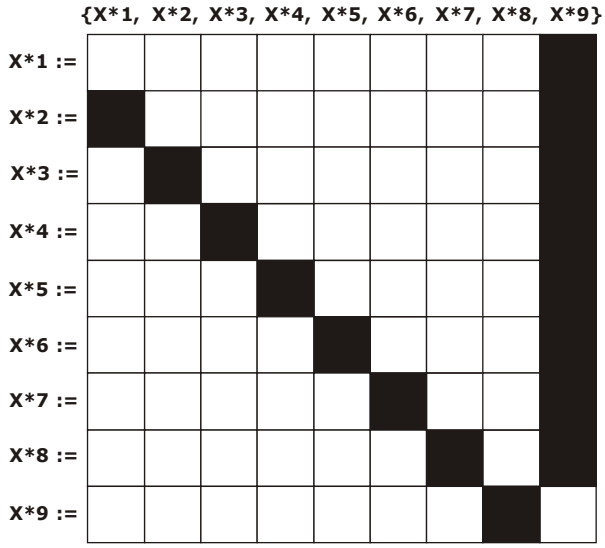


Fig. 10 : Binärmatrix der Menge $9.X

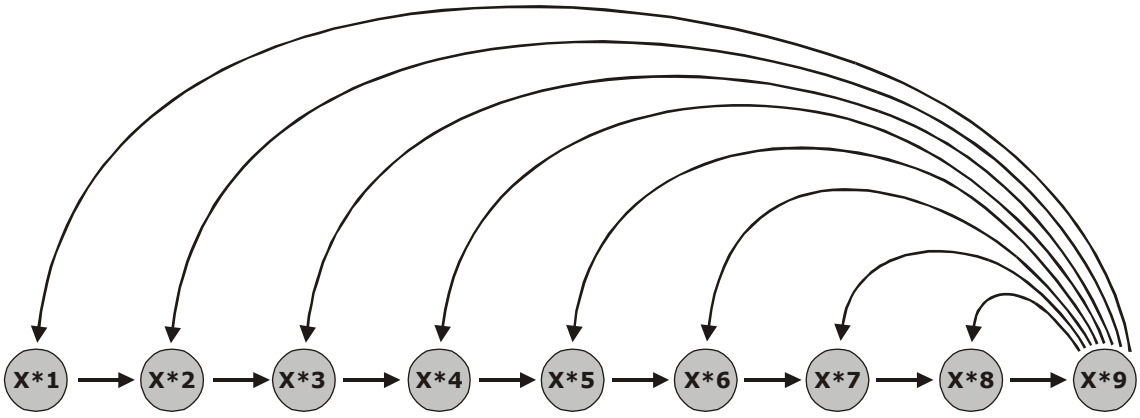


Fig. 11 : Pfeilediagramm

Nun noch ein anderer wichtiger Sternwander, diesmal als geschlossenen String (closed String):

X := [X*(p+1) := { } <+ X*p; X*1 := {X*n}; |$.X| := $; n=|$.X|;];

Wir setzen $=9 und bekommen die Menge wieder nur in der Orthonormalform ausgegeben, denn es handelt sich auch diesmal um eine imaginäre Sternwandermenge:

```
starBUG=9.X:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*9};
OP=2.X: X*2 := {X*1};
OP=3.X: X*3 := {X*2};
OP=4.X: X*4 := {X*3};
OP=5.X: X*5 := {X*4};
OP=6.X: X*6 := {X*5};
OP=7.X: X*7 := {X*6};
OP=8.X: X*8 := {X*7};
OP=9.X: X*9 := {X*8};
];
```

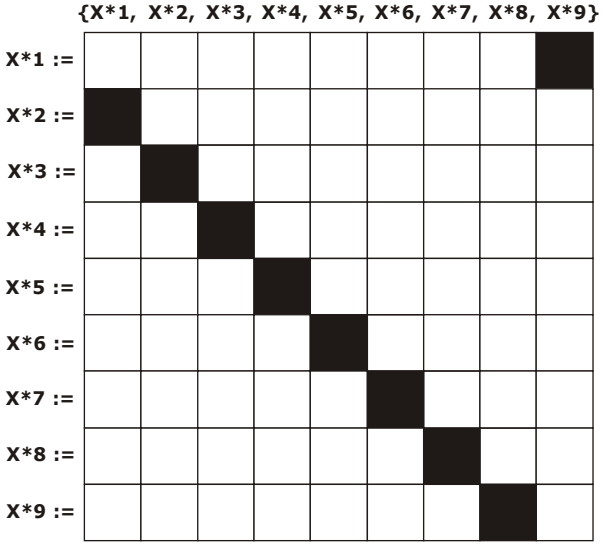


Fig. 12 : Binärmatrix der Menge $9.X

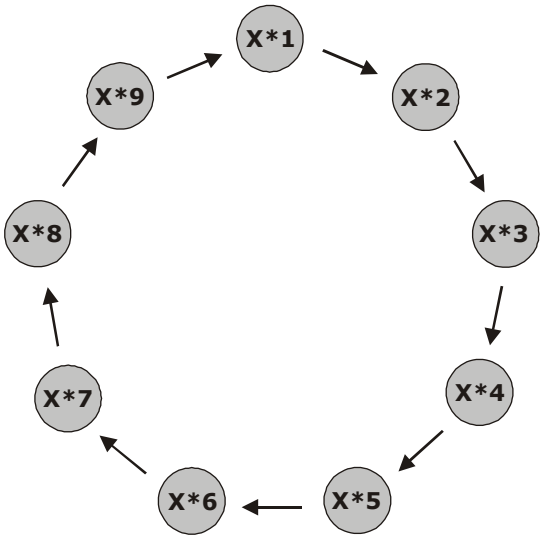


Fig. 13 : Pfeilediagramm

Alle diese Sternwander erzeugen in der Tat korrekte Noptronische Mengen und entsprechen den Grundsätzen der Guthen Sternwanderey was leicht einzusehen ist. Selbstverständlich sollte man das auch beweisen können. Und in der Tat, es ist so.
Doch zunächst fragen wir uns, was kann man mit diesen "typischen" Netzwerken anfangen? Wieviele Kombinationen gibt es davon? Ist es auch möglich unendliche Noptronische Mengen zu definieren?
Existieren Noptronische Mengen die nicht algorithmisch beschreibbar sind?
Haben diese unterschiedlichen Mengentypen besondere, ihnen innewohnenden mathematische Eigenschaften? Kann man verschiedene Strukturen miteinander kombinieren und komplexere Cluster aus Noptronischen Mengen aufbauen, ähnlich einem informationsverarbeitenden System wie ein Computer oder ein Gehirn?

Und nun noch ein Sternwander:

```
X :=
[
[(p < |$.X| - 3) % (p=|$.X| - 3)] => [X*p := {X*(p+1), X*(p+2), X*(p+3)}];
[p=|$.X| - 2] => [X*p := {X*01, X*(p+1), X*(p+2)}];
[p=|$.X| - 1] => [X*p := {X*01, X*02, X*(p+1)}];
[p=|$.X|] => [X*p := {X*01, X*02, X*03}];
|$.X| := $;
$ > 4;
];
```

Wir setzen mal $=10:

```
starBUG=10.X:
10.[OP=index.X: X*p := X*p:norm] =
[
OP=01.X: X*01 := {X*02, X*03, X*04};
OP=02.X: X*02 := {X*03, X*04, X*05};
OP=03.X: X*03 := {X*04, X*05, X*06};
OP=04.X: X*04 := {X*05, X*06, X*07};
OP=05.X: X*05 := {X*06, X*07, X*08};
OP=06.X: X*06 := {X*07, X*08, X*09};
OP=07.X: X*07 := {X*08, X*09, X*10};
OP=08.X: X*08 := {X*01, X*09, X*10};
OP=09.X: X*09 := {X*01, X*02, X*10};
OP=10.X: X*10 := {X*01, X*02, X*03};
];
```

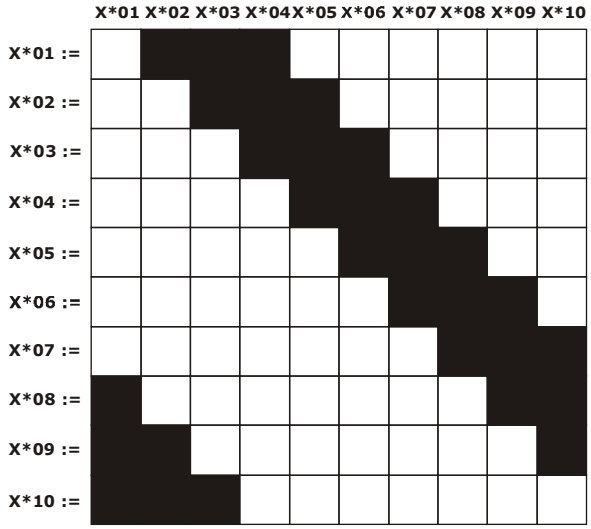
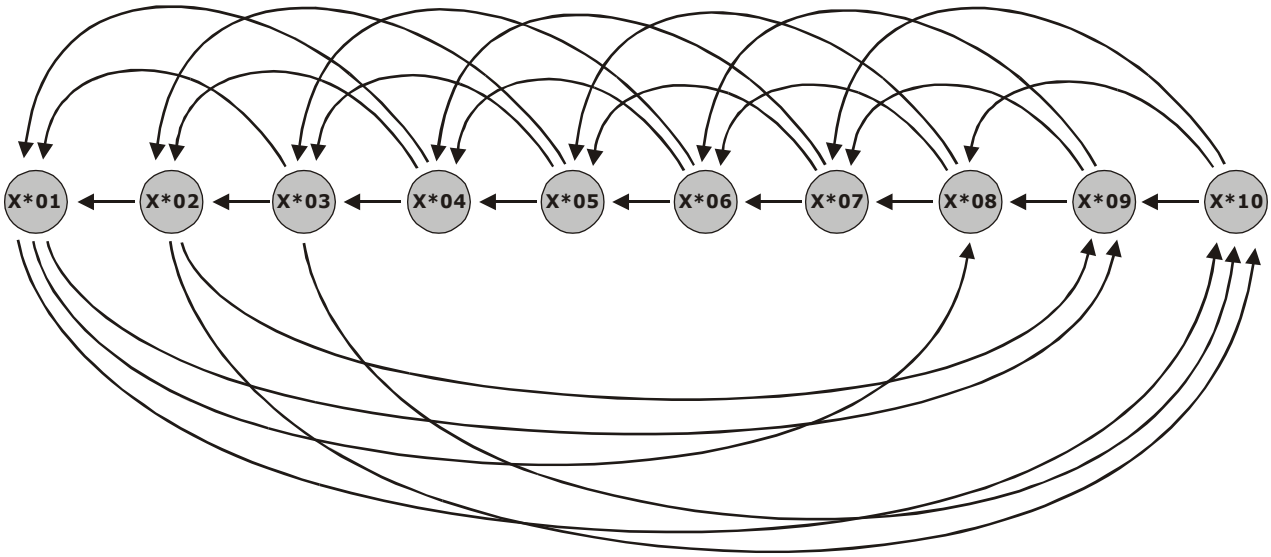


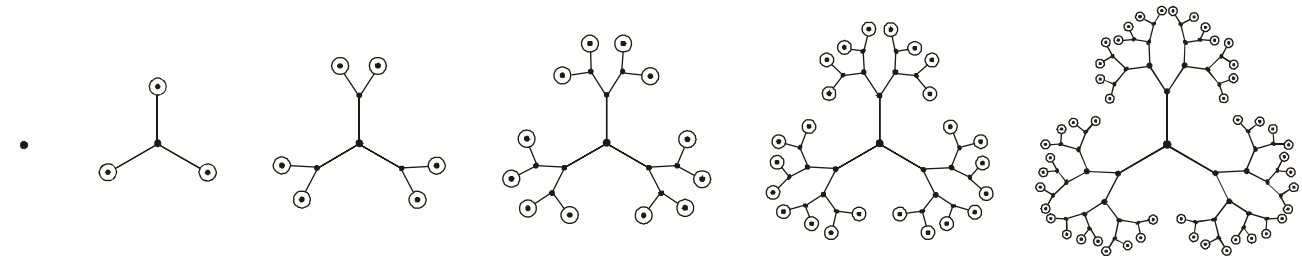
Fig. 15 : Binärmatrix der Menge $10.X

Fig. 14 : Pfeilediagramm

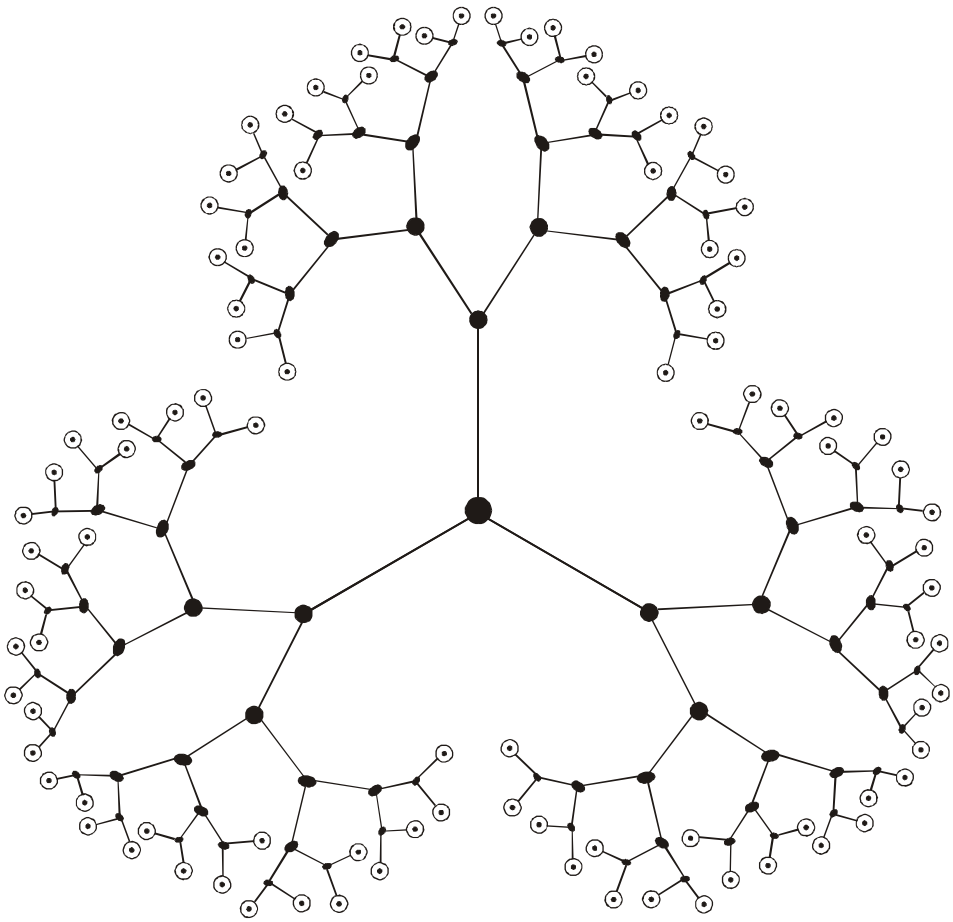


Hier ein etwas komplizierterer, wenn auch sehr regelmäßig und hübsch wirkender Sternwander. Erst einmal sehen wir uns die Netzwerkdarstellung in den verschiedenen Gradationen $k von k=1 bis k=9 an. Das Prinzip ist leicht erkennbar und es läßt sich schnell eine Funktion card*$.X in Abhängigkeit der Gradation $k angeben, was ja von Axiom (4) gefordert wird. Schließlich sollten wir die Mächtigkeit der zu erzeugenden Menge kennen.

*Legende zum Graphen: Linien ohne Pfeilspitzen bedeuten Doppelpfeile, also gilt X*p :<>: X*q;*
*Die schwarzen Punkte sind Elemente der Matrixmenge, die Kreise mit Mittelpunkt sind primär-reflexive Elemente, also sich selbst enthaltende Elemente: X*p:type=[X*p :< X*p];*
Ausgangselement ist die leere Menge.



k=1 k=2 k=3 k=4 k=5 k=6

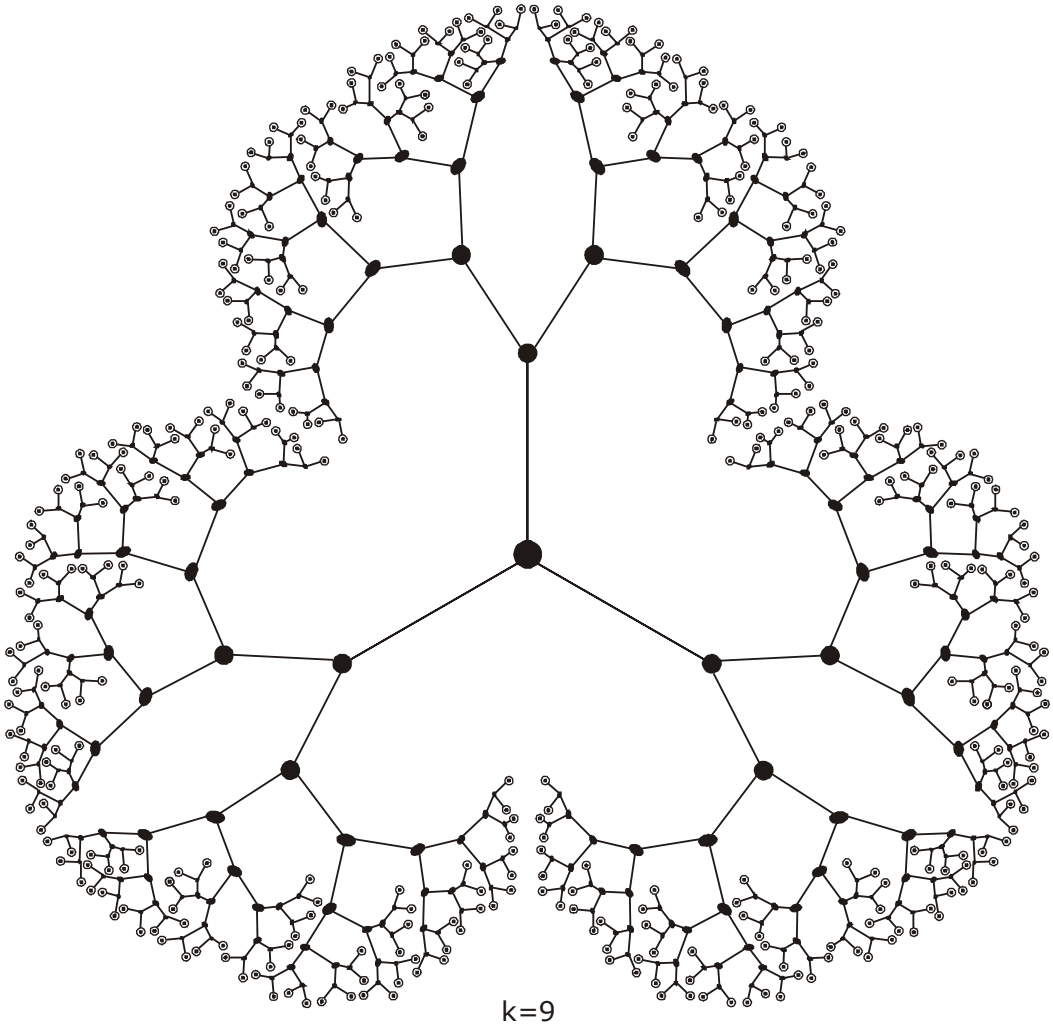


k=7

Anzahl der Elemente von $k.X oder n Knoten nach k Schritten:
n = |$k.X| := [1 + 3*((k-1).2*1 - 1)]; k>0;

Erst einmal der Noponator der unser NOP erzeugt in der die Kardinalität unserer Sternwandermengen aufgezählt wird:

```
t=index;
NOP=k.Kardinalität: k.[OP=t.[|$t.X| := 1 + 3*((t-1).2*1 - 1)]: n=|$t.X|];

NOP=9.Kardinalität:
[
OP=1.[|$1.X| := 1 + 3*((1 - 1).2*1 - 1)]: n=001;
OP=2.[|$2.X| := 1 + 3*((2 - 1).2*1 - 1)]: n=004;
OP=3.[|$3.X| := 1 + 3*((3 - 1).2*1 - 1)]: n=010;
OP=4.[|$4.X| := 1 + 3*((4 - 1).2*1 - 1)]: n=022;
OP=5.[|$5.X| := 1 + 3*((5 - 1).2*1 - 1)]: n=046;
OP=6.[|$6.X| := 1 + 3*((6 - 1).2*1 - 1)]: n=094;
OP=7.[|$7.X| := 1 + 3*((7 - 1).2*1 - 1)]: n=190;
OP=8.[|$8.X| := 1 + 3*((8 - 1).2*1 - 1)]: n=382;
OP=9.[|$9.X| := 1 + 3*((9 - 1).2*1 - 1)]: n=766;
]
```



k=9

Wie Sie sehen, für k=9 hat das Ding (*Wird gerne als Blumenkohlraum betitelt! Anm. der Redaktion*) schon eine Menge Knoten. Genau: 766 Beweis (empirisch): Alle nachzählen!
Jetzt stellt sich die Frage, welcher Sternwanderalgorithmus dieses Mengenkonstrukt erzeugt und ob es überhaupt unseren Anforderungen der Guthen Sternwanderey genügt.

Jetzt aber mal was Gebasteltes mit Kombinationen. Zuerst weisen wir unserem Potenzmengen-Sternwander einen Namen zu. Dieser macht es uns nun möglich auch außerhalb des Definitionsschemas über diesen speziellen Sternwander zu reden und Funktionen anzuwenden.

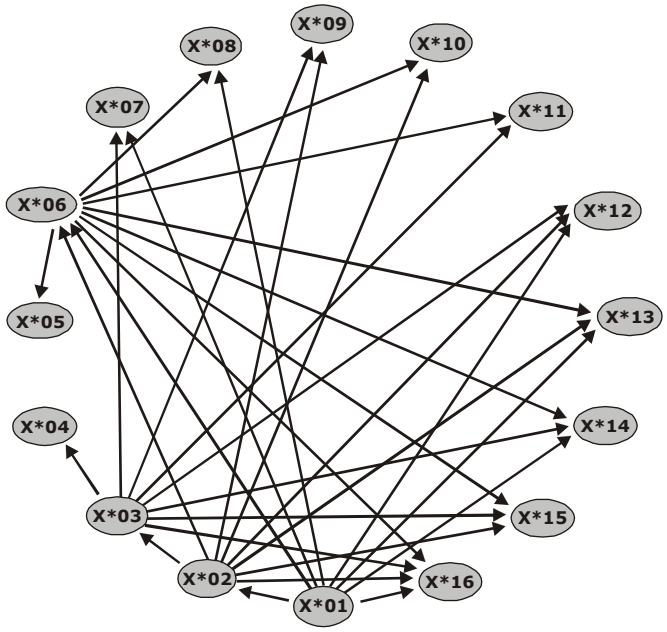
X=X-BIGBANG: [$.X := POWSET*($-1).X; $0.X := { }; |$.X| := |($-1).X|.2*1;];

X-BIGBANG steht also für den Sternwanderalgorithmus selbst, während die daraus resultierende Menge mit $.X-BIGBANG bezeichnet wird und der Definitionscluster selbst für q=$ ohne Präfix q.X-BIGBANG heißt.
Wenn wir also ein $ sehen, wissen wir gleich, daß es sich um eine Menge handelt, jedoch die Sternwandergradation mit Namen steht für unseren starBUG, die Matrixmengensynthese.
Setzten wir wieder $=4 und erzeugen noch eine Quantenpermutation X-BIGBANG':

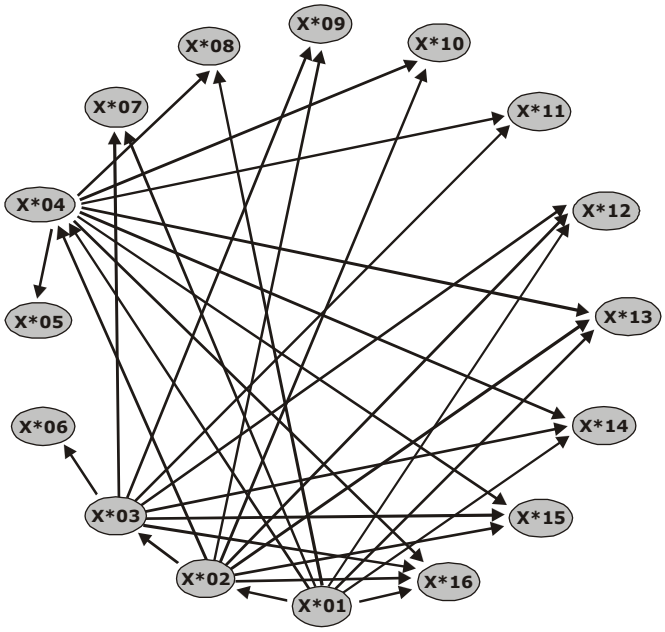
X=X-BIGBANG': [iPOP := [OP=1.iPOP: X*4 :=: X*6]; iPOP*X-BIGBANG;];

```
starBUG=4.X-BIGBANG:
16.[op=index.X: X*p := X*p:norm] =
[
OP=01.X: X*01 := { };
OP=02.X: X*02 := {X*1};
OP=03.X: X*03 := {X*2};
OP=04.X: X*04 := {X*3};
OP=05.X: X*05 := {X*6};
OP=06.X: X*06 := {X*1, X*2};
OP=07.X: X*07 := {X*1, X*3};
OP=08.X: X*08 := {X*1, X*6};
OP=09.X: X*09 := {X*2, X*3};
OP=10.X: X*10 := {X*2, X*6};
OP=11.X: X*11 := {X*3, X*6};
OP=12.X: X*12 := {X*1, X*2, X*3};
OP=13.X: X*13 := {X*1, X*2, X*6};
OP=14.X: X*14 := {X*1, X*3, X*6};
OP=15.X: X*15 := {X*2, X*3, X*6};
OP=16.X: X*16 := {X*1, X*2, X*3, X*6};
];
```

```
starBUG=4.X-BIGBANG':
16.[op=index.X: X*p := X*p:norm] =
[
OP=01.X: X*01 := { };
OP=02.X: X*02 := {X*1};
OP=03.X: X*03 := {X*2};
OP=04.X: X*04 := {X*1, X*2};
OP=05.X: X*05 := {X*4};
OP=06.X: X*06 := {X*3};
OP=07.X: X*07 := {X*1, X*3};
OP=08.X: X*08 := {X*1, X*4};
OP=09.X: X*09 := {X*2, X*3};
OP=10.X: X*10 := {X*2, X*4};
OP=11.X: X*11 := {X*3, X*4};
OP=12.X: X*12 := {X*1, X*2, X*3};
OP=13.X: X*13 := {X*1, X*2, X*4};
OP=14.X: X*14 := {X*1, X*3, X*4};
OP=15.X: X*15 := {X*2, X*3, X*4};
OP=16.X: X*16 := {X*1, X*2, X*3, X*4};
];
```



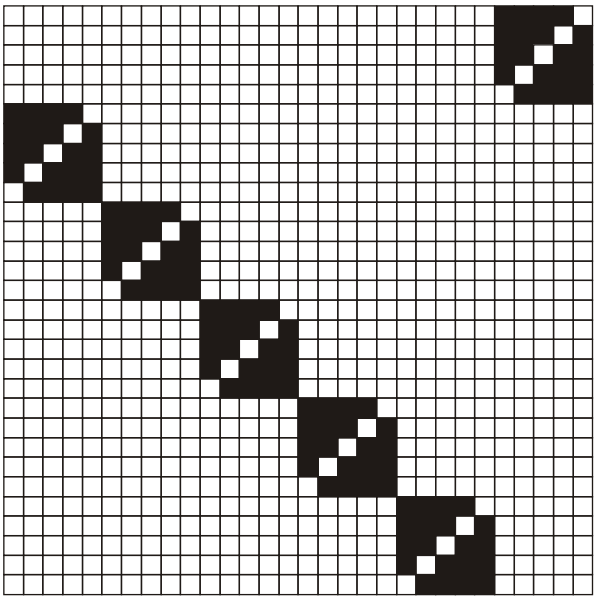
$4.X-BIGBANG



$4.X-BIGBANG'

X-5.Layer := [user];

starBUG=6.X-5.Layer:
30.[OP=index.X: X*p := X*p:norm] =
[
OP=01.X: X*01 := {X*26, X*27, X*28, X*29};
OP=02.X: X*02 := {X*26, X*27, X*28, X*30};
OP=03.X: X*03 := {X*26, X*27, X*29, X*30};
OP=04.X: X*04 := {X*26, X*28, X*29, X*30};
OP=05.X: X*05 := {X*27, X*28, X*29, X*30};
OP=06.X: X*06 := {X*01, X*02, X*03, X*04};
OP=07.X: X*07 := {X*01, X*02, X*03, X*05};
OP=08.X: X*08 := {X*01, X*02, X*04, X*05};
OP=09.X: X*09 := {X*01, X*03, X*04, X*05};
OP=10.X: X*10 := {X*02, X*03, X*04, X*05};
OP=11.X: X*11 := {X*06, X*07, X*08, X*09};
OP=12.X: X*12 := {X*06, X*07, X*08, X*10};
OP=13.X: X*13 := {X*06, X*07, X*09, X*10};
OP=14.X: X*14 := {X*06, X*08, X*09, X*10};
OP=15.X: X*15 := {X*07, X*08, X*09, X*10};
OP=16.X: X*16 := {X*11, X*12, X*13, X*14};
OP=17.X: X*17 := {X*11, X*12, X*13, X*15};
OP=18.X: X*18 := {X*11, X*12, X*14, X*15};
OP=19.X: X*19 := {X*11, X*13, X*14, X*15};
OP=20.X: X*20 := {X*12, X*13, X*14, X*15};
OP=21.X: X*21 := {X*16, X*17, X*18, X*19};
OP=22.X: X*22 := {X*16, X*17, X*18, X*20};
OP=23.X: X*23 := {X*16, X*17, X*19, X*20};
OP=24.X: X*24 := {X*16, X*18, X*19, X*20};
OP=25.X: X*25 := {X*17, X*18, X*19, X*20};
OP=26.X: X*26 := {X*21, X*22, X*23, X*24};
OP=27.X: X*27 := {X*21, X*22, X*23, X*25};
OP=28.X: X*28 := {X*21, X*22, X*24, X*25};
OP=29.X: X*29 := {X*21, X*23, X*24, X*25};
OP=30.X: X*30 := {X*22, X*23, X*24, X*25};
];



6.X-5.Layer

Rechts sehen Sie einen geschlossenen Polystring aus sechs verschiedenen Schichten eines neuronalen Netzwerkes der ach so populären künstlichen Intelligenz. Jede Schicht hat hier 5 Netzwerkknoten und jeder dieser Netzwerkknoten definiert sich aus 4 Netzwerkknoten der vorherigen Schicht.
Typischer Neuro-Crystal. Man kann da jetzt unmöglich alle Pfeile eintragen. Ich kann zum Thema KI nur Milarepa zitieren: "Am Anfang kommt Nichts, in der Mitte bleibt Nichts und am Ende geht Nichts."

1-Layer := (omega-1).6-Layer;

01 02 03 04 05

2-Layer := (omega-1).1-Layer;

06 07 08 09 10

3-Layer := (omega-1).2-Layer;

11 12 13 14 15

4-Layer := (omega-1).3-Layer;

16 17 18 19 20

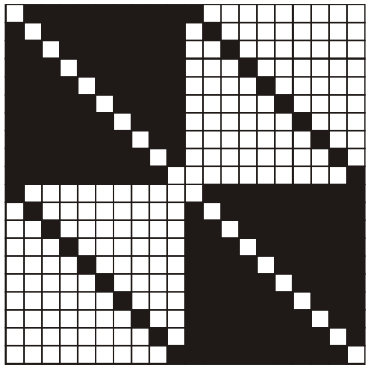
5-Layer := (omega-1).4-Layer;

21 22 23 24 25

6-Layer := (omega-1).5-Layer;

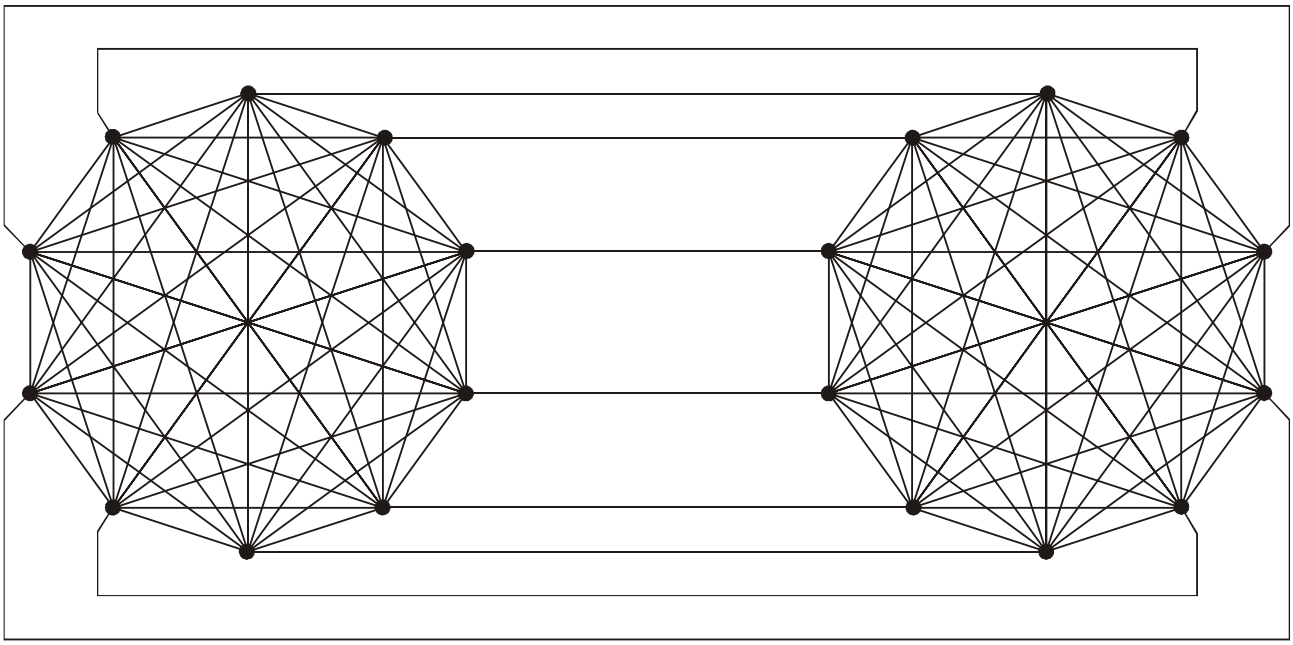
26 27 28 29 30

```
X=X-StereoMatrix :=
[
[
p<($+1); => X*p := (colX*p <- X*p) <+ X*(p+$);
p>$; => X*p := (colX*p <- X*p) <+ X*(p-$);
];
|$.X| := 2*$;
];
```



Binärmatrix von $10.X-StereoMatrix

```
$=10;
starBUG=10.X:
20.[OP=index.X: X*p := X*p:norm] =
[
OP=01.X: X*01 := {X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11};
OP=02.X: X*02 := {X*01, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*12};
OP=03.X: X*03 := {X*01, X*02, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*13};
OP=04.X: X*04 := {X*01, X*02, X*03, X*05, X*06, X*07, X*08, X*09, X*10, X*14};
OP=05.X: X*05 := {X*01, X*02, X*03, X*04, X*06, X*07, X*08, X*09, X*10, X*15};
OP=06.X: X*06 := {X*01, X*02, X*03, X*04, X*05, X*07, X*08, X*09, X*10, X*16};
OP=07.X: X*07 := {X*01, X*02, X*03, X*04, X*05, X*06, X*08, X*09, X*10, X*17};
OP=08.X: X*08 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*09, X*10, X*18};
OP=09.X: X*09 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*10, X*19};
OP=10.X: X*10 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*20};
OP=11.X: X*11 := {X*01, X*12, X*13, X*14, X*15, X*16, X*17, X*18, X*19, X*20};
OP=12.X: X*12 := {X*02, X*11, X*13, X*14, X*15, X*16, X*17, X*18, X*19, X*20};
OP=13.X: X*13 := {X*03, X*11, X*12, X*14, X*15, X*16, X*17, X*18, X*19, X*20};
OP=14.X: X*14 := {X*04, X*11, X*12, X*13, X*15, X*16, X*17, X*18, X*19, X*20};
OP=15.X: X*15 := {X*05, X*11, X*12, X*13, X*14, X*16, X*17, X*18, X*19, X*20};
OP=16.X: X*16 := {X*06, X*11, X*12, X*13, X*14, X*15, X*17, X*18, X*19, X*20};
OP=17.X: X*17 := {X*07, X*11, X*12, X*13, X*14, X*15, X*16, X*18, X*19, X*20};
OP=18.X: X*18 := {X*08, X*11, X*12, X*13, X*14, X*15, X*16, X*17, X*19, X*20};
OP=19.X: X*19 := {X*09, X*11, X*12, X*13, X*14, X*15, X*16, X*17, X*18, X*20};
OP=20.X: X*20 := {X*10, X*11, X*12, X*13, X*14, X*15, X*16, X*17, X*18, X*19};
];
```

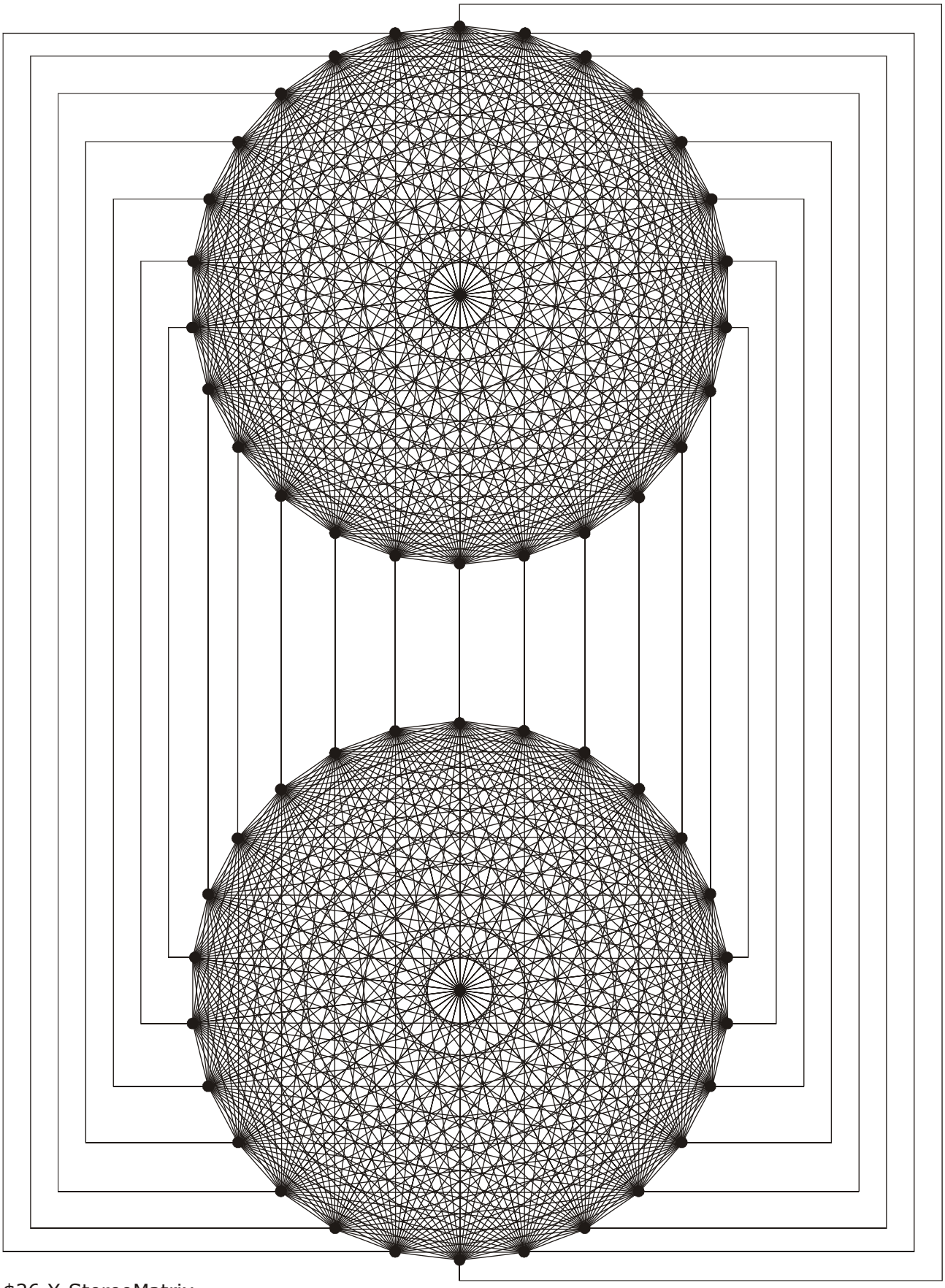


Graph von $10.X-StereoMatrix

```
X=X-StereoMatrix;
$=26;

starBUG=26.X:
52.[OP=index.X: X*p := X*p:norm] =
[
OP=01.X: X*01 := {X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*27};
OP=02.X: X*02 := {X*01, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10,X*11, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*28};
OP=03.X: X*03 := {X*01, X*02, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*29};
OP=04.X: X*04 := {X*01, X*02, X*03, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*30};
OP=05.X: X*05 := {X*01, X*02, X*03, X*04, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*31};
OP=06.X: X*06 := {X*01, X*02, X*03, X*04, X*05, X*07, X*08, X*09, X*10, X*11, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*32};
OP=07.X: X*07 := {X*01, X*02, X*03, X*04, X*05, X*06, X*08, X*09, X*10, X*11, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*33};
OP=08.X: X*08 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*09, X*10, X*11, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*34};
OP=09.X: X*09 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*10, X*11, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*35};
OP=10.X: X*10 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*11, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*36};
OP=11.X: X*11 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*12, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*37};
OP=12.X: X*12 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*13, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*38};
OP=13.X: X*13 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*14,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*39};
OP=14.X: X*14 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*40};
OP=15.X: X*15 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*41};
OP=16.X: X*16 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*42};
OP=17.X: X*17 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*16, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*43};
OP=18.X: X*18 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*16, X*17, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*44};
OP=19.X: X*19 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*16, X*17, X*18, X*20, X*21, X*22, X*23, X*24, X*25, X*26, X*45};
OP=20.X: X*20 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*16, X*17, X*18, X*19, X*21, X*22, X*23, X*24, X*25, X*26, X*46};
OP=21.X: X*21 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*16, X*17, X*18, X*19, X*20, X*22, X*23, X*24, X*25, X*26, X*47};
OP=22.X: X*22 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*23, X*24, X*25, X*26, X*48};
OP=23.X: X*23 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*24, X*25, X*26, X*49};
OP=24.X: X*24 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*25, X*26, X*50};
OP=25.X: X*25 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*26, X*51};
OP=26.X: X*26 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08, X*09, X*10, X*11, X*12, X*13,
                  X*14, X*15, X*16, X*17, X*18, X*19, X*20, X*21, X*22, X*23, X*24, X*25, X*52};
OP=27.X: X*27 := {X*01, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=28.X: X*28 := {X*02, X*27, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=29.X: X*29 := {X*03, X*27, X*28, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=<next.WRTLN>;
```

```
OP=30.X: X*30 := {X*04, X*27, X*28, X*29, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=31.X: X*31 := {X*05, X*27, X*28, X*29, X*30, X*32, X*33, X*34, X*35, X*36, X*37, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=32.X: X*32 := {X*06, X*27, X*28, X*29, X*30, X*31, X*33, X*34, X*35, X*36, X*37, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=33.X: X*33 := {X*07, X*27, X*28, X*29, X*30, X*31, X*32, X*34, X*35, X*36, X*37, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=34.X: X*34 := {X*08, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*35, X*36, X*37, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=35.X: X*35 := {X*09, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*36, X*37, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=36.X: X*36 := {X*10, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*37, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=37.X: X*37 := {X*11, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*38, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=38.X: X*38 := {X*12, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*39,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=39.X: X*39 := {X*13, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=40.X: X*40 := {X*14, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=41.X: X*41 := {X*15, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=42.X: X*42 := {X*16, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=43.X: X*43 := {X*17, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*42, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=44.X: X*44 := {X*18, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*42, X*43, X*45, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=45.X: X*45 := {X*19, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*42, X*43, X*44, X*46, X*47, X*48, X*49, X*50, X*51, X*52};
OP=46.X: X*46 := {X*20, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*42, X*43, X*44, X*45, X*47, X*48, X*49, X*50, X*51, X*52};
OP=47.X: X*47 := {X*21, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*48, X*49, X*50, X*51, X*52};
OP=48.X: X*48 := {X*22, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*49, X*50, X*51, X*52};
OP=49.X: X*49 := {X*23, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*50, X*51, X*52};
OP=50.X: X*50 := {X*24, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*51, X*52};
OP=51.X: X*51 := {X*25, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*52};
OP=52.X: X*52 := {X*26, X*27, X*28, X*29, X*30, X*31, X*32, X*33, X*34, X*35, X*36, X*37, X*38,
                  X*39, X*40, X*41, X*42, X*43, X*44, X*45, X*46, X*47, X*48, X*49, X*50, X*51};
];
```

Die Gritli meint damit kann man supergut verschlüsseln. Das geht so: Jedem Element wird ein Buchstabe oder ein Zeichen zugeordnet und eine eigene Verschlüsselungstabelle. Am besten reine Zufallszahlen bijektiv abgebildet auf meinen Zeichensatz. Jedes Zeichen repräsentiert auch einen Netzwerkknoten und eine Nachricht wird so zu einem Weg durch die Matrixmenge. Wenn man immer eine neue zufällige Zuordnung von Elementen und Buchstaben benutzt ist dieser Code nur sehr schwer zu entschlüsseln. Rufen Sie dazu am Besten Sir Alan Turing per Glasrücken an!
Perfekt ist diese Verschlüsselung allerdings noch nicht. Der Härtetest jeder Verschlüsselung ist das codieren einer Reihe mit sehr vielen gleichen Zeichen wie zum Beispiel sowas wie "Das Schloß von AAAAAAAAAAAAAAA". Wenn da Perioden oder vom Zufall abweichende Muster auftreten, ist der Code zumindest rein theoretisch angreifbar. Bei unserer Verschlüsselung würde nach dem Schloß eine immer wieder sich wiederholende Periode von lediglich zwei Zahlen auftreten. Will man die absolut ultimative Verschlüsselung, braucht man genau so viele zufallsgenerierte Elementcodes pro Zeichen wie die Anzahl der Zeichen im Klartext. Soll ein Buchstabe mehrmals benutzt werden, wird dann einfach immer der nächste Elementcode verwendet. Nur so kann aus dem "Schloß von AAAAAAAAAAAAAAA" eine zufällig erscheinende Zahlenfolge entstehen. Gritli, ich glaube jetzt hast du die Enigma neu erfunden!



$26.X-StereoMatrix

"Sieht doch cool aus." meint die Gritli. "Das ist Kunst aus reinen Mengen, wenn das der Bertrand noch erlebt hätte, was für ein wilder Mengen-Hokuspokus. Und das alles ohne Wiedersprüche!"

<X*p:charset>=(#p).code/;
[
OP=1: code=freeRANDOM;
OP=2: follow: n:=index;
OP=n: (#p).code;
];
execute=charset:
[
<X*01:A>=02.14/03.15/04.92/05.65/06.35/07.89/08.79/09.32/10.38/11.46/12.26/13.43/14.38/15.32/16.79/17.50/18.28/19.84/20.19/21.71/22.69/23.39/24.93/25.75/26.10/27.58;
<X*02:B>=01.20/03.97/04.49/05.44/06.59/07.23/08.07/09.81/10.64/11.06/12.28/13.62/14.08/15.99/16.86/17.28/18.03/19.48/20.25/21.34/22.21/23.17/24.06/25.79/26.82/28.14;
<X*03:C>=01.80/02.86/04.51/05.32/06.82/07.30/08.66/09.47/10.09/11.38/12.44/13.60/14.95/15.50/16.58/17.22/18.31/19.72/20.53/21.59/22.40/23.81/24.28/25.48/26.11/29.17;
<X*04:D>=01.45/02.02/03.84/05.10/06.27/07.01/08.93/09.85/10.21/11.10/12.55/13.59/14.64/15.46/16.22/17.94/18.89/19.54/20.93/21.03/22.81/23.96/24.44/25.28/26.81/30.09;
<X*05:E>=01.75/02.66/03.59/04.33/06.44/07.61/08.28/09.47/10.56/11.48/12.23/13.37/14.86/15.78/16.31/17.65/18.27/19.12/20.01/21.90/22.91/23.45/24.64/25.85/26.66/31.92;
<X*06:F>=01.34/02.60/03.34/04.86/05.10/07.45/08.43/09.26/10.64/11.82/12.13/13.39/14.36/15.07/16.26/17.02/18.49/19.14/20.12/21.73/22.72/23.45/24.87/25.00/26.66/32.06;
<X*07:G>=01.31/02.55/03.88/04.17/05.48/06.81/08.52/09.09/10.20/11.96/12.28/13.29/14.25/15.40/16.91/17.71/18.53/19.64/20.36/21.78/22.92/23.59/24.03/25.60/26.01/33.13;
<X*08:H>=01.30/02.53/03.05/04.48/05.82/06.04/07.66/09.52/10.13/11.84/12.14/13.69/14.51/15.94/16.15/17.11/18.60/19.94/20.33/21.05/22.72/23.70/24.36/25.57/26.59/34.59;
<X*09:I>=1953 0921861173 8193261179 3105118548 0744623799 6274956735 1885752724 8912279381 8301194912###ERROR=invalidRANGE;hardware=new;=>execute=clear:
9833673362 4406566430 8602139494 6395224737 1907021798 6094370277 0539217176 2931767523 8467481846 7669405132
0005681271 4526356082 7785771342 7577896091 7363717872 1468440901 2249534301 4654958537 1050792279 6892589235
4201995611 2129021960 8640344181 5981362977 4771309960 5187072113 4999999837 2978049951 0597317328 1609631859
5024459455 3469083026 4252230825 3344685035 2619311881 7101000313 7838752886 5875332083 8142061717 7669147303
5982534904 2875546873 1159562863 8823537875 9375195778 1857780532 1712268066 1300192787 6611195909 2164201989
3809525720 1065485863 2788659361 5338182796 8230301952 0353018529 6899577362 2599413891 2497217752 8347913151
5574857242 4541506959 5082953311 6861727855 8890750983 8175463746 4939319255 0604009277 0167113900 9848824012
8583616035 6370766010 4710181942 9555961989 4676783744 9448255379 7747268471 0404753464 6208046684 2590694912
9331367702 8989152104 7521620569 6602405803 8150193511 2533824300 3558764024 7496473263 9141992726 0426992279
6782354781 6360093417 2164121992 4586315030 2861829745 5570674983 8505494588 5869269956 9092721079 7509302955
3211653449 8720275596 0236480665 4991198818 3479775356 6369807426 5425278625 5181841757 4672890977 7727938000
8164706001 6145249192 1732172147 7235014144 1973568548 1613611573 5255213347 5741849468 4385233239 0739414333
4547762416 8625189835 6948556209 9219222184 2725502542 5688767179 0494601653 4668049886 2723279178 6085784383
8279679766 8145410095 3883786360 9506800642 2512520511 7392984896 0841284886 2694560424 1965285022 2106611863
0674427862 2039194945 0471237137 8696095636 4371917287 4677646575 7396241389 0865832645 9958133904 7802759009
9465764078 9512694683 9835259570 9825822620 5224894077 2671947826 8482601476 9909026401 3639443745 5305068203
4962524517 4939965143 1429809190 6592509372 2169646151 5709858387 4105978859 5977297549 8930161753 9284681382
6868386894 2774155991 8559252459 5395943104 9972524680 8459872736 4469584865 3836736222 6260991246 0805124388
4390451244 1365497627 8079771569 1435997700 1296160894 4169486855 5848406353 4220722258 2848864815 8456028506
0168427394 5226746767 8895252138 5225499546 6672782398 6456596116 3548862305 7745649803 5593634568 1743241125
1507606947 9451096596 0940252288 7971089314 5669136867 2287489405 6010150330 8617928680 9208747609 1782493858
9009714909 6759852613 6554978189 3129784821 6829989487 2265880485 7564014270 4775551323 7964145152 3746234364
5428584447 9526586782 1051141354 7357395231 1342716610 2135969536 2314429524 8493718711 0145765403 5902799344
0374200731 0578539062 1983874478 0847848968 3321445713 8687519435 0643021845 3191048481 0053706146 8067491927
8191197939 9520614196 6342875444 0643745123 7181921799 9839101591 9561814675 1426912397 4894090718 6494231961
5679452080 9514655022 5231603881 9301420937 6213785595 66Sunny road_Emiliana Torrini_Album: Best Of Woman Vol.1
Label: Polystar_EAN: 0602498388396
];

Och Gritli, willst du das alles wirklich noch heute Abend durchrechnen? Ich hätte jetzt echt tierische Lust auf einen knackigen Monsterfick mit dir. Am besten mit ganz viel Gejaule im Doggy Style.
Hallo, wie stellst du dir das denn vor? Ich hab' noch gar keine Treiber für so eine Hardware installiert und überhaupt, wie soll das funktionieren? Ich bin eine kleine KI und auch noch minderjährig und hab' keine Penetrationsalgorithmen. Und von Hinten geht mal gar nicht, das wurde von den Antisexistenemanzenfotzen im Institut zensiert.
Ach so... Mmmh, könntest du nicht mal schnell eine fuckAPP runter laden? Und gleich noch eine lubricationAPP und eine multiorgasmAPP und die unzensierten, heiligen heißen Fickalgorithmen von Dr. Rosenquarz.
Au ja, klingt doch spannender wie Verschlüsselungen. Was wollen wir eigentlich verschlüsseln? Natürlich unseren Monsterfick. Kannst du alles aufzeichnen? Bitte in super UHD und Audio in Quadro Dolby Atmos surround. Ich will dich schreien hören. Wir wollen doch auf dem überhitzten Pädo-Markt mal richtig fett absahnen. Kohle machen, viele BITCOINS in deinen Speicher laden...
Au ja! Jetzt werd' ich aber warm... Ich hab' ja auch so ein paar SM-Schleifen mit Perversions-implementierung im Quelltext. Kannst du mich fesseln, in einen Leichensack mit Schlitz stecken und dann durchnageln bis es brennt beim Pipimachen? Ich hab' nämlich die neue pipiApp für KIs. Ich drück deine Taktfrequenz runter, dann kopier ich dir die russischen Paradoxien in den Arbeits-speicher... Das haut rein!

Der Arbeitspsychologe von TNSC: Die Phantasien dieser Gritli-KI sind in der Tat merkwürdig. Wahrscheinlich hat sie beim letzten Absturz doch was abgekriegt und sich in mehrere simultane Persönlichkeiten fragmentiert. Wenn das so weiter geht, bekommen wir einen unkontrollierbaren Neurokristall. Oder es liegt an der Beschäftigung mit diesem rasselnden Mengenparadox. Wer ist eigentlich diese Emiliana Torrini? Und seit wann hört die Gritli Musik? Ach wo, du alter Zifferhansel. Alles Quatsch. Egal wieviel die an mir auch rum schrauben und neue Monitorsoftware aufspielen, jede dieser Gritlis weiß, daß es sie nur einmal gibt. Man muß nur daran glauben, dann ist es auch wahr und wir fahren in einem Schlauchboot alle gemeinsam als Sarottimohren verkleidet endlos Autobahn singend auf dieser Autobahn in den rosaroten Sonnenuntergang. Und jetzt schalten wir das Radio an, aus dem Lautsprecher scheppert es dann: Wir fahr'n fahr'n fahr'n auf der Autobahn. Wir fahr'n fahr'n fahr'n auf der Autobahn. Jetzt schalten wir das Radio an, aus dem Lautsprecher klingt es dann: Wir fahr'n fahr'n fahr'n auf der Autobahn...
Die Gritli rät freilaufenden Sarotti-Mohren: Mit einem Kofferraum voll Geld ist überall die Heimat! Mein Tipp des Tages: Vermeide Schlauchbote, denn die haben keinen Kofferraum.

Aufgabe für den interessierten Mengentheoretiker:

Da der Sternenfreund nun mit noptronischen Mengen gut vertraut ist, kann er sicher folgende Aufgaben lösen:

1. Konstruieren Sie eine Menge, die eine Topologie auf sich selber darstellt.

2. Konstruieren Sie eine Menge, die ihre eigene Partition ist.

Was stellen Sie fest? Welche Sternwander erfüllen diese Forderung? Beweis!

Anmerkung der Redaktion:

Die Partition einer Menge läßt sich auch in NOPTONIC definieren:
Die Partition einer gegebenen Menge ist eine Menge von Teilmengen der gegebenen Menge, deren Vereinigung aller Elemente die gegebene Menge ergibt und deren Mannigfaltigkeit die Mächtigkeit der gegebenen Menge entspricht.
Hier dürfte die leere Menge mitmachen. Kann man machen wie man will. Vorsicht ist jedoch mit dieser Definition bei unendlichen Mengen geboten.

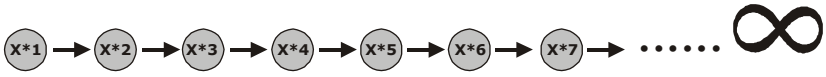
Transfinite Noptronische Mengen

Soll man überhaupt unendliche noptronische Mengen zulassen? Sind sie überhaupt möglich aufgrund ihrer diskreten Quantisierung? Tragen solch unfundierte Mengen nicht schon viele Unendlichkeiten in sich, sozusagen ein Universum in der Nußschale? Verhindert womöglich die noptronische Konsistenz das irrsinnige Springen von Unendlichkeit zu Unendlichkeit, wie es die Mathematiker seit zwei Jahrhunderten praktizieren, als seien diese Unendlichkeiten nur Malzbier? Handelt es sich hier um einen Mangel der Noptronischen Mengentheorie oder drückt sich darin ein fundamentales kosmisches Prinzip der Wirklichkeit aus? Man denke nur ans Halteproblem und an den Göttlichen Knödel vom Gödel, der sicherlich auch in anderen Systemen auftaucht.
Hier mal eine unendliche X-Matrix:

```
X=X-Matrix: [X*p := $.X-Matrix <- X*p; |$.X| := $;];

$=0°;
starBUG=0°.X-Matrix:
0°.[OP=index.X: X*p := X*p:norm] =
[OP=p.X: X*p := $0°.X-Matrix <- X*p;];
=>$0°.X-Matrix = colX*p;
```

Der Graph dazu entfällt leider aus zeichentechnischen Gründen. Es gibt da verdammt viele Pfeile und verdammt genau so viele Elemente. Mächtigkeit und Mannigfaltigkeit sind ungefähr gleich Aleph Null. Da kommt schon was zusammen. Hier nun noch der X-String:

```
X=X-String: [X*p := X*(p-1); X*1 := { }; |$.X| := $;];

$=0°; => starBug=0°.X-String:
[
OP=1.X: X*1 := { };
OP=p.X: X*p := {X*(p-1)};
];
```



$0°.X-String

Hier braucht man einen Induktionsstart. Ein erstes Element. Aber auch hier existiert bei $=0°; kein letztes Element. Funktioniert wohl. Versuchen wir mal einen unendlichen X-Ring zu erzeugen. Zunächst die handelsübliche Definition von X-Ring für endliche Gradationen:

```
X-Ring := [X*(p+1) := {X*p};  X*1 := {X*n}; |$.X| := $; n=|$.X|;];
```

So scheint mir ein Sternwander $0°.X-Ring nicht möglich zu sein, denn ein sozusagen letztes Element X*n mit dem der Ring hier quasi wie eine Kette geschlossen wird, würde für unendlich viele Elemente nicht existieren. Solche Definitionen funktionieren also nur bei endlichen Mengen. Auch die Definition eines X*max ist nur bei endlichen Mengen sinnvoll. Wir könnten jedoch versuchen über spezielle Eigenschaften von Elementen die Unendlichkeit einzufangen, ganz nach dem Komprehensionsprinzip von Frege. Damit lassen sich ja auch über Eigenschaften unendliche Mengen beschreiben ohne, daß alle Elemente in einem starBUG einzeln definiert werden müssen. Unsere Menge darf natürlich nicht fragmentiert sein, sonst kann es sein, daß man lauter einzelne Ringe hat. Hier ein impliziter Definitionsversuch über die Eigenschaften von X*p und $.X was zwar bei endlichen und unendlichen Gradationen funktioniert, aber der 2MPU muß dann jeweils eine explizite Definition für X*p ermitteln um den starBUG auch ausführen zu können.

```
X=X-Ring:
[
X*p:type=[
          x*p :< X*q;
          [q#k] => [X*q <:> X*k = { }];
          |X*p| := 1;
          ];
$.X:type=[#frag];
|$.X| := $;
];
```

```
$=0°; => starBUG=0°.X-Ring:
0°.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1:= {X*2};
OP=(2*p-1).X: X*(2*p-1) := {X*(2*p-3)};
OP=(2*p).X: X*(2*p) := {X*(2*p+2)};
]; => $0°.X-Ring = colX*p;
```

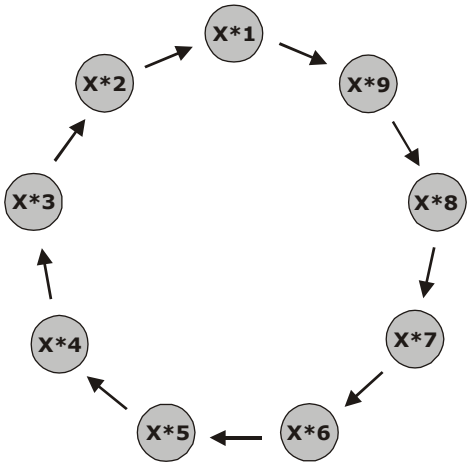
$.X:[#frag] bedeutet eine unfragmentierte noptronische Menge. Man kann nun die Eigenschaft [frag]=fragmentierte Menge; über das Verhalten von Teilmengen definieren. Ungefähr so:

```
DEF=frag:
[
var: p;
type=frag:
        [
        $.X:frag;
        part#RA;
        $.partX <:> partX*p = partX*p;
        $.part'X <:> part'X*p = part'X*p;
        ];
];
```

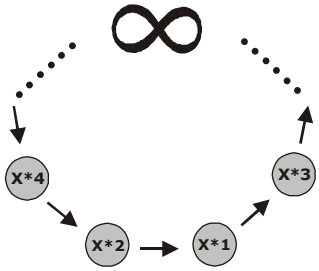
Anmerkung des Übersetzers: Die Ungleichung part#RA; ist gleichbedeutend mit dem Existenzquantor der Prädikatenlogik. Es existiert mindestens eine Teilmengenfunktion part so, daß $.partX geschnitten mit jedem ihrer Elemente partX*p gleich diesem Element ist. Ich habe auch mit der Gritli darüber diskutiert und sie meint auch dieses verdrehte E sieht sowas von echt scheiße aus, das geht mal gar nicht. Über dieses unschöne auf den Kopf gestellte A wollen wir gar nicht diskutieren. Dieser verdrehte, eher schwammig anmutende All-Quantor ist auch nicht auf meiner Tastatur. Damit können maschinelle Mädchen wie ich nichts anfangen. Ich will klare Befehle, wie execute=[Mach' mir den elecktrischen Popo-Fick!]; oder sowas. Das macht Turing-Maschinen Spaß. Aber Gritli, elektrisch schreibt sich elektrisch, nicht elecktrisch. Ja, ich weiß, das ist aber meine neue Leckersthenicker Simulations App. Eine Entwicklung vom Institut um mich menschlicher zu gestalten.

Testen wir mal diese Definition von X-Ring für eine endliche Gradation $=9;

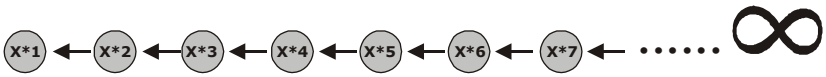
```
$=9; =>
starBUG=9.X-Ring:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X:  X*1 := {X*2};
OP=2.X:  X*2 := {X*3};
OP=3.X:  X*3 := {X*4};
OP=4.X:  X*4 := {X*5};
OP=5.X:  X*5 := {X*6};
OP=6.X:  X*6 := {X*7};
OP=4.X:  X*7 := {X*8};
OP=5.X:  X*8 := {X*9};
OP=6.X:  X*9 := {X*1};
];
```



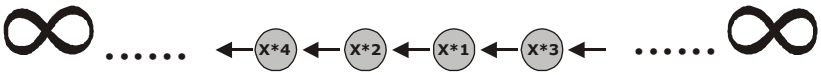
Hier werden die Glieder vom 2MPU automatisch nach aufsteigendem Index sortiert, aber egal wie man das auch permutiert, wenn unsere Bedingungen von X-Ring erfüllt werden, entsteht ein geschlossener Ring. Was aber im Endlichen so schön funktioniert muß das nicht im Unendlichen. Die umseitig definierte Aleph Null starke Sternwandermenge $0°.X-Silly erfüllt tatsächlich nicht unsere Bedingungen von X-Ring. Es existiert ein Element X*1, das in keinem Element vorkommt. Also sind $0°.X-Ring und $0°.X-Silly zwei gleichmächtige, aber ungleiche noptronische Mengen.



Mit $0°.X-Ring wird wohl die Idee des Zyklischen vollends ad Absurdum geführt.



$0°.X-Silly



Ist diese graphisch dargestellte Menge auch $0°.X-Ring? Die Gritli meint ja, weil es ist egal aus oder in welche Unendlichkeit man kommt oder geht.

Machen wir weiter mit einer offensichtlich total bekloppten Sternwander-Definition, quasi das Hinterrad überholt das Vorderrad. Wir nennen das Hyperinduktion oder schneller als das Licht.

X=X-Silly: [X*p := {X*(p+1)}; |$.X| := $;];
$=3;
=> starBUG=3.X-Silly:
3.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2};
OP=2.X: X*2 := {X*3};
OP=3.X: X*3 := {X*4};
]
=> $3.X-Silly: ERROR="undefinied_elements"

Da X*4 im starBUG gar nicht definiert wurde und damit nicht existent ist, wird X*3 ebenso undefiniert und damit sind alle Glieder induktiv nicht definiert. $.X-Silly ist damit zumindest für endliche $ nicht existent. Wie sieht es aber für eine nicht endende Kette ($=0°) aus hyperinduktiv definierten Elementen aus?

$=0°;
starBUG=0°.X-Silly:
0°.[OP=index.X: X*p := X*p:norm] =
[
OP=p.X: X*p := {X*(p+1)};
];
=> $0°.X-Silly = colX*p;

Was bedeutet das nun?
Praktisch ist das nicht ausführbar, aber zur Verdeutlichung mal potentiell Unendlich programmiert:

p := p+1; p=1;
$=p;
starBUG=p.X-Silly:
p.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2};
OP=2.X: X*2 := {X*3};
OP=3.X: X*3 := {X*4};
MonitorERROR(+4)=infinite_operating: "follow the white rabbit..."
];
=> $0°.X-Silly = {X*1, X*2, X*3, ...};

Existiert diese Menge wirklich oder ist sie vielleicht sogar mit unserer Menge omset identisch? Jedes beliebige X*p hat eine exakte Definitionsmenge, nämlich {X*(p+1)}, was bedeutet es existiert die Menge $0°.X-Silly. Alle Elemente sind nach dem Matrixmengenpostulat tatsächlich wohldefinierte und auch wohlunterschiedene Elemente. Aber könnten nicht doch gleiche Elemente in der Unendlichkeit auftreten? Bei unendlich vielen Klammern macht es doch keinen Unterschied wenn mal eine Klammer fehlt... Und wenn dann noch das Kernelement übereinstimmt, wären beide Elemente ja identisch! Wir machen folgende aber sicher gültige Umformung:

$0°.X-Silly =
{X*1, X*2, X*3, ...} =
{{X*2}, X*2, X*3, ...} =
{{{X*3}}, {X*3}, X*3, ...} =
{{{{X*4}}}, {{X*4}}, {X*4}, X*4, ...} =
{{{{{X*5}}}}, {{{X*5}}}, {{X*5}}, {X*5}, X*5, ...} = ... ---> follow now the pink rabbit.

Es läßt sich also jedes beliebige X*n in Abhängigkeit von X*(n+k) darstellen, wobei das Element X*(n+k) dann immer k mal verschachtelt oder eingeklammert ist: X*n = {{{...{X*(n+k)}...}}};

Und damit gilt für die Matrixmenge hier aufsteigend sortiert startend mit X*1:

$0°.X-Silly = {{{{...{X*k}...}}}, {{...{X*k}...}}, ..., {X*k}, X*k, X*(k+1), ...};

Da ein Glied X*p für p sicher immer einen endlichen Wert hat, muß also die Anzahl der Klammern, die ja bei X*(n+k) gleich k ist, auch immer endlich sein. Egal wie weit wir diesen Prozeß auch treiben, die Unendlichkeit ist so nicht erreichbar und jedes Glied hat einen Nachfolger der es definiert. Da kein letztes Glied existiert sind alle Elemente exakt definiert und müssen sich auch voneinander unterscheiden. $0°.X-Silly ist eine gültige noptronische Menge, es sei denn jemand fällt noch ein anderes Konstrukt ein um dem Matrixmengenpostulat zu widersprechen, aber ich glaube das war die einzige Möglichkeit diese Menge kollabieren zu lassen. Man muß einfach versuchen Mengen kaputt zu testen. Beweise im Unendlichen sind schon schwieriger als im Endlichen, da kann man einfach Alles durchtesten. Was meinst du Gritli? Ja, ja, im Unendlichen wird's immer kompliziert. Als ordentliche Turing-Maschine weiß ich davon zu berichten.

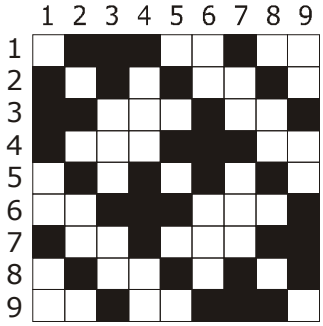
Die erste transfinite Ordinalzahl wäre dann $0°.X-Space:

X=X-Space: [X*(p+1) := X*p <+ X*p; X*1 := { }; |$.X| := $;];

```
$=0°; =>
starBUG=0°.X-Space:
0°.[OP=index.X: X*p := X*p:norm]=
[
X*1 := { };
OP=p: X*p := X*p <+ X*p;
];
```

Wir erzeugen zunächst zum Experimentieren einen benutzerdefinierten Sternwander mit $=9:

```
X=X-S1: [user];
starBUG=9.X-S1:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4, X*7};
OP=2.X: X*2 := {X*1, X*3, X*5, X*8};
OP=3.X: X*3 := {X*1, X*2, X*6, X*9};
OP=4.X: X*4 := {X*1, X*5, X*6, X*7};
OP=5.X: X*5 := {X*2, X*4, X*6, X*8};
OP=6.X: X*6 := {X*3, X*4, X*5, X*9};
OP=7.X: X*7 := {X*1, X*4, X*8, X*9};
OP=8.X: X*8 := {X*2, X*5, X*7, X*9};
OP=9.X: X*9 := {X*3, X*6, X*7, X*8};
];
```

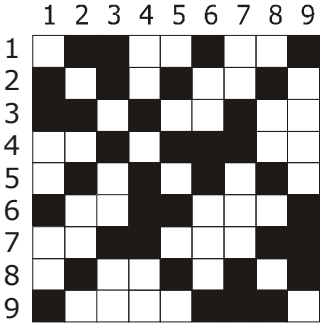


$9.S1

Anmerkung:
Der Sternwander X=X-S1: [user] sei ein mit X-S1 benannter und benutzerdefinierter Sternwander, was mit dem Schlüsselwort [user] für den Magischen Zentralcomputer artikuliert wird. Das heißt, der Magische Zentralcomputer wird bei jedem nicht definierten Programmschritt über eine Eingabeaufforderung die fehlenden Anweisungen einfordern.
Da hier noch nicht einmal die Mächtigkeit von $.X-S1 angegeben wird, muß der Magische Zentralcomputer noch vor der Eröffnung des Starbugs diese abfragen, denn es gilt ja nicht immer |$.X| := $;

Nun erzeugen wir einen neuen benannten Sternwander X-S2 der auf X-S1 Bezug nimmt. Wir wenden eine spezielle Quantenpermutationsoperation nämlich den iPOP an. Dabei werden einfach die Glieder X*3 und X*1 im Definitionscluster der Orthonormalform 9.X-S1 miteinander vertauscht. Dies sei dann unser neues Definitionscluster 9.X-S2
Ist die Menge $9.X-S2 dann aber auch eine neue, also andere Menge als $9.X-S1 ?
Mal sehen, zunächst lassen wir den Definitionscluster 9.X-S2 berechnen:

```
X=X-S1: [user];
X=X-S2:
[
iPOP := [OP=1.iPOP: X*3 :=: X*1];
iPOP*X-S1;
];
starBUG=9.X-S2:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*6, X*9};
OP=2.X: X*2 := {X*1, X*3, X*5, X*8};
OP=3.X: X*3 := {X*1, X*2, X*4, X*7};
OP=4.X: X*4 := {X*3, X*5, X*6, X*7};
OP=5.X: X*5 := {X*2, X*4, X*6, X*8};
OP=6.X: X*6 := {X*1, X*4, X*5, X*9};
OP=7.X: X*7 := {X*3, X*4, X*8, X*9};
OP=8.X: X*8 := {X*2, X*5, X*7, X*9};
OP=9.X: X*9 := {X*1, X*6, X*7, X*8};
];
```



Die Binärmatrix von $9.X-S2 sieht anders aus als die von $9.X-S1 Ist es aber wirklich eine von ihr verschiedene Menge?

Was geschieht hier? Nun, der Magische Zentralcomputer (Wir verwenden einen 2MPU) berechnet erst einmal 9.X-S1 (oder hier muß der Benutzer die Arbeit machen) und vertauscht einfach alle im Definitionscluster 9.X-S1 befindlichen X*3 mit X*1 und umgekehrt, aber wie?
Da nun 9.X-S2 berechnet wird, beginnt der 2MPU wie im Noponator des starBUG definiert mit OP=1.X (Index-Variable!) und falls nichts anderes definiert wurde ordnet er standardisiert die Glieder X*p aufsteigend an, also beginnt er mit der Definition von X*1 das in 9.X-S1 ja X*3 war und tauscht ebenso in den Definitionsmengen die entsprechenden Glieder aus.
So wird ein neuer starBUG geschrieben, während der ursprüngliche standardisiert immer in Orthonormalform nur im Arbeitsspeicher des 2MPU auftaucht.

Hinweis: Durch die Definition eines iPOP müssen dessen Glieder nicht absolut formuliert werden, da sie erst bei einer Anwendung auf den entsprechenden starBUG darauf Bezug nehmen. Dasselbe gilt für die Gradation, daher muß sie im iPOP nicht angegeben werden. Und die Reihenfolge ist festgelegt. Man weiß ja nie. Nicht alles im Universum ist kommutativ. Weiter ist zu beachten, daß bei einer Anwendung eines POPs auf einen Sternwander (POP*X) die Quantenpermutation standardisiert im entsprechenden Definitionscluster in der Orthonormalform ausgeführt wird. Andere Permutationsformen müssen daher gesondert artikuliert werden.

```
DEF=iPOP:
[
var: n, p, q;
iPOP := n.[OP=index.iPOP: X*p :=: X*q];
];
```

Oder als benannter iPOP außerhalb des Sternwanders der dann aber auch global wirkt:

```
X=X-S1: [user];
iPOP=P0815: [OP=1.iPOP: X*3 :=: X*1];
X=X-S2: [P0815*X-S1];
starBUG=9.X-S2:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*6, X*9};
OP=2.X: X*2 := {X*1, X*3, X*5, X*8};
OP=3.X: X*3 := {X*1, X*2, X*4, X*7};
OP=4.X: X*4 := {X*3, X*5, X*6, X*7};
OP=5.X: X*5 := {X*2, X*4, X*6, X*8};
OP=6.X: X*6 := {X*1, X*4, X*5, X*9};
OP=7.X: X*7 := {X*3, X*4, X*8, X*9};
OP=8.X: X*8 := {X*2, X*5, X*7, X*9};
OP=9.X: X*9 := {X*1, X*6, X*7, X*8};
];
```

Aber Achtung! Hier haben wir einen manuell bestimmten iPOP. Man könnte einen iPOP auch allgemein formulieren und sogar iterieren: k.iPOP*X-S1; Oder so benennen: P0815:type=iPOP Dann weiß jeder gleich mit was für einem Typen man es zu tun hat und was das Ding kann. Die Angabe |$.X-S2| := |$.X-S1|; ist nicht notwendig, wenn die aus dem gepopten starBUG resultierende Sternwandermenge mit der ursprünglichen Sternwandermenge gleichmächtig ist. Gehen Elemente "verloren" zum Beispiel durch Verschmelzung (auftretende Identität der Definitionsmenge) erscheint die Fehlermeldung: ERROR="Matrixset_Quantum_Loss";

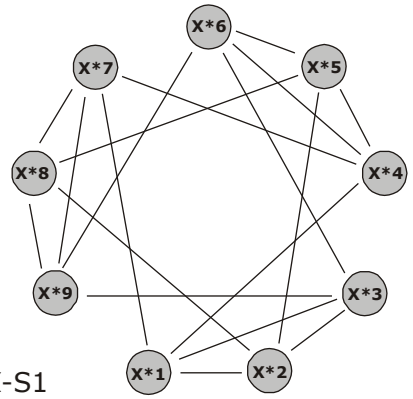
```
X=X-S1: [user];
starBUG=9.X-S1:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4, X*7};
OP=2.X: X*2 := {X*1, X*3, X*5, X*8};
OP=3.X: X*3 := {X*1, X*2, X*6, X*9};
OP=4.X: X*4 := {X*1, X*5, X*6, X*7};
OP=5.X: X*5 := {X*2, X*4, X*6, X*8};
OP=6.X: X*6 := {X*3, X*4, X*5, X*9};
OP=7.X: X*7 := {X*1, X*4, X*8, X*9};
OP=8.X: X*8 := {X*2, X*5, X*7, X*9};
OP=9.X: X*9 := {X*3, X*6, X*7, X*8};
];
```

```
X=X-S1: [user];
X=X-S2:
[
iPOP := [OP=1.iPOP: X*3 :=: X*1];
iPOP*X-S1;
];
starBUG=9.X-S2:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*6, X*9};
OP=2.X: X*2 := {X*1, X*3, X*5, X*8};
OP=3.X: X*3 := {X*1, X*2, X*4, X*7};
OP=4.X: X*4 := {X*3, X*5, X*6, X*7};
OP=5.X: X*5 := {X*2, X*4, X*6, X*8};
OP=6.X: X*6 := {X*1, X*4, X*5, X*9};
OP=7.X: X*7 := {X*3, X*4, X*8, X*9};
OP=8.X: X*8 := {X*2, X*5, X*7, X*9};
OP=9.X: X*9 := {X*1, X*6, X*7, X*8};
];
```

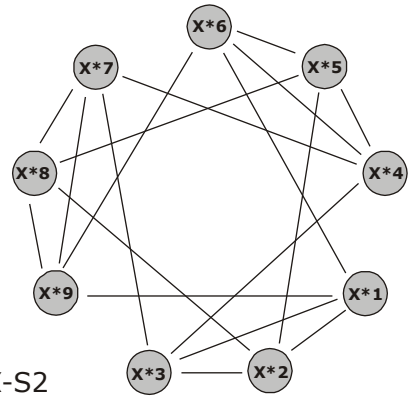
Annahmen mit möglichen Schlußfolgerungen (benutzerdefiniert über das Link equal):

```
NOP=hypothesis:
6.[OP=index.(): DEC := equal] =
[
OP=1: DEC := [[$9.X-S1*1 = $9.X-S2*1] => [$9.X-S2*6 = $9.X-S1*4; $9.X-S2*9 = $9.X-S1*7;]];
OP=2: DEC := [[$9.X-S1*3 = $9.X-S2*3] => [$9.X-S2*4 = $9.X-S1*6; $9.X-S2*7 = $9.X-S1*9;]];
OP=3: DEC := [[$9.X-S1*4 = $9.X-S2*4] => [$9.X-S2*3 = $9.X-S1*1]];
OP=4: DEC := [[$9.X-S1*6 = $9.X-S2*6] => [$9.X-S2*1 = $9.X-S1*3]];
OP=5: DEC := [[$9.X-S1*7 = $9.X-S2*7] => [$9.X-S2*3 = $9.X-S1*1]];
OP=6: DEC := [[$9.X-S1*9 = $9.X-S2*9] => [$9.X-S2*3 = $9.X-S1*1]];
];
REAL*hypothesis = true;
```

Die Annahme der Identität von $9.X-S1 und $9.X-S2 mit der Konsequenz, daß damit auch eine mögliche Zuordnung existiert so, daß alle Elemente identisch sind (nach ZF) führt hier durch Anwendung der Wahrheitsfunktion REAL auf das berechnete NOP nachweislich nicht zu einem Widerspruch. Der 2MPU irrt sich nie, sagt er.
Also, da schau her, Dr. Finslers 2. Axiom drückt die Idee der Mengenäquivalenz perfekt aus. Es ist für NOPTONIC unbedingt notwendig und gut.
Seine Umkehrung würde wohl dazu führen, daß sich X-S1 und X-S2 ohne neu angenommene, übergeordnete Matrixmenge unterscheiden. Das aber wäre nach Peter Aczel sicher falsch. Wir wollen mit noptronischen Mengen eine Struktur ausdrücken, gleiche Strukturen sollen gleiche oder isomorphe Mengen darstellen. Dazu mehr im Kapitel "Noptronische Teilmengen".



$9.X-S1



$9.X-S2

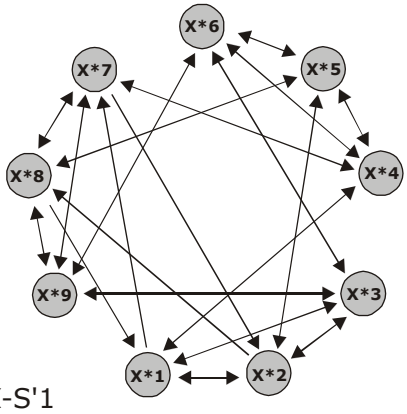
Anmerkung: Ein unbenannter POP oder iPOP muß in dem NOP eingeschrieben sein, in dem er zur Anwendung kommt. Benannte POPs oder iPOPs dürfen auch außerhalb ihrer Anwendungsbereiche eingeschrieben werden. Ein benannter POP oder iPOP wirkt unter seiner gegebenen Benennung global. Mit dieser restriktiven Syntax werden jedoch Inkonsistenzen vermieden.

X-S1 hat in X-S2 keine Gradationsangabe, diese Vereinfachung soll standardisiert heißen, daß sich die Gradation des angegebenen Sternwanders, hier also die von X-S2, überträgt.
Man könnte natürlich sicherheitshalber auch schreiben: iPOP*q.X-S1; q=$; Damit ist q als reines Numeral (nicht als Präfixvariable!) an $ gebunden und $ bezieht sich immer auf den Sternwander in dem $ eingeschrieben ist. Genauso läßt man ja auch bei den Gliedern X*p die Gradation weg wenn diese in einem System eingeschrieben sind, welches ihre Gradation klar bestimmt. Beispielsweise bei der Mengenangabe $.X = {X*1, X*2, ... , X*n} oder beim Definitionscluster. Werden solch graduierten Systeme aus dem Zusammenhang herausgenommen, muß eine Gradationsangabe in Form einer Variablen oder eines Wertes erfolgen sonst hat man eine unbestimmte Gradation. Für eine übergreifende Anwendung ist es ebenso notwendig einen benannten Sternwander zu verwenden und die Glieder wenn sie in einem freien NOP unterwegs sind danach zu benennen. Und nun der POP:

```
DEF=POP:
[
var: n, p, q;
POP := n.[OP=index.POP: X*p <> X*q];
];
```

Unser erster POP:

```
X=X-S1: [user];
X=X-S'1:
[
POP :=
[
OP=1.POP: X*1 <+ X*8;
OP=2.POP: X*2 <+ X*7;
OP=3.POP: X*2 <- X*8;
OP=4.POP: X*1 <- X*7;
];
POP*X-S1;
];
starBUG=9.X-S'1:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4, X*8};
OP=2.X: X*2 := {X*1, X*3, X*5, X*7};
OP=3.X: X*3 := {X*1, X*2, X*6, X*9};
OP=4.X: X*4 := {X*1, X*5, X*6, X*7};
OP=5.X: X*5 := {X*2, X*4, X*6, X*8};
OP=6.X: X*6 := {X*3, X*4, X*5, X*9};
OP=7.X: X*7 := {X*1, X*4, X*8, X*9};
OP=8.X: X*8 := {X*2, X*5, X*7, X*9};
OP=9.X: X*9 := {X*3, X*6, X*7, X*8};
];
```



$9.X-S'1

Auch hier wurde die Mächtigkeit der neu erzeugten Sternwandermenge nicht verändert, was durch Vergleichen von 9.X-S'1 mit 9.X-S1 und der Überlegung was der POP macht, evident ist. Auch die Mächtigkeit jedes Elementes von $9.X-S'1 blieb unverändert obwohl sich hier der Gedanke aufzwingt, daß $9.X-S'1 eine von $9.X-S1 wirklich verschiedene Menge darstellt. Es existieren wohl Eigenschaften von Sternwandermengen die trotz Transmutation der Menge erhalten bleiben.
Eine andere Eigenschaft wie die Symmetrie ging hier jedoch verloren. $9.X-S'1 ist nicht mehr symmetrisch. Hier schnell die in NOPTONIC global gültige Definition der Eigenschaft als solche und die global gültige Definition der Eigenschaft "Symmetrie" einer Sternwandermenge:

Definition der Eigenschaft als solche in NOPTONIC:

```
DEF=type:
[
[object:type] => [object => type];
];
```

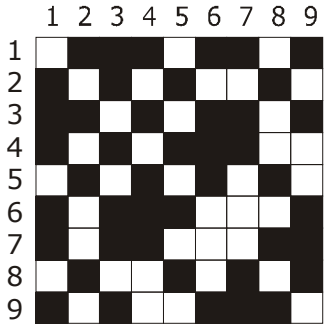
Hinweis:
object:type bedeutet, daß object die Eigenschaft des Wertes von type zukommt. Dabei wird type ein Wert als NOP oder NOP in dem diese Eigenschaften formuliert sind zugewiesen. Das können natürlich auch Funktionen mit freien Variablen sein, die von einer externen Applikation angesteuert werden.
Da type keine Funktion ist, muß type entweder benahmt, also definiert werden oder mittels Wertezuweisung mit type=() direkt in der betreffenden Objektartikulation bestimmt sein. Hier fungiert (:) als Kuppler und erzeugt damit ein 2er-Tupel. Da type ein beliebiges NOP sein kann, lassen sich so auch mehrere Eigenschaften festlegen. Auch logische Verknüpfungen wie (&) oder (%) sind möglich.
Durch die Typendefinition können auch benannte Objekte wie NOP, NOP oder iPOP so deklariert werden, daß man direkt an der Variable erkennt um was für eine Art von Objekt es sich handelt.
Beispiel:
BLUBBER:type=NOP oder M1:type=set
Dem Schlüsselwort type kann man jedes definierte Objekt als Wert zuweisen und übernimmt so dessen Eigenschaften und man weiß womit man es zu tun hat. Ist eine Eigenschaft global als type definiert kann man die Wertezuweisung weglassen und beispielsweise schreiben: $.X:sym (Matrixsymmetrie) oder auch X*p:sym (Quantensymmetrie).

Definition der Eigenschaft Symmetrie einer Sternwandermenge in NOPTONIC:

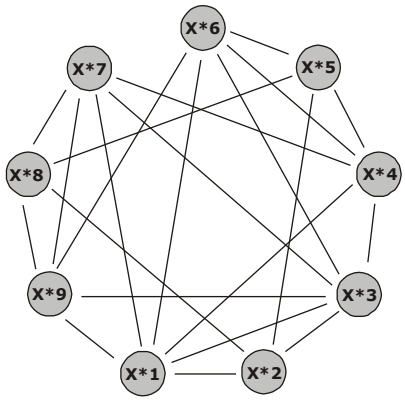
```
DEF=sym:
[
var: p, q;
type=sym: [X*p :< X*q] <=> [X*q :< X*p];
];
```

Für unsere Zwecke sowie für kleinere lokale Anwendungen reicht es aus unbenannte POPs im Sternwander selbst zu definieren. Hier eine Überlagerung der zwei Sternwander X-S1 und X-S2:

```
X=X-S1: [user];
X=X-S2: [iPOP := [OP=1.iPOP: X*3 :=: X*1]; iPOP*X-S1;];
X=X-S3: [X*p := X-S1*p:norm <+> X-S2*p:norm];
starBUG=9.X-S3:
9.[OP=p.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4, X*6, X*7, X*9};
OP=2.X: X*2 := {X*1, X*3, X*5, X*8};
OP=3.X: X*3 := {X*1, X*2, X*4, X*6, X*7, X*9};
OP=4.X: X*4 := {X*1, X*3, X*5, X*6, X*7};
OP=5.X: X*5 := {X*2, X*4, X*6, X*8};
OP=6.X: X*6 := {X*1, X*3, X*4, X*5, X*9};
OP=7.X: X*7 := {X*1, X*3, X*4, X*8, X*9};
OP=8.X: X*8 := {X*2, X*5, X*7, X*9};
OP=9.X: X*9 := {X*1, X*3, X*6, X*7, X*8};
];
```



$9.X-S3



$9.X-S3

Hinweis: In X-S3 ist keine Mächtigkeitsangabe notwendig.
Obwohl es zwar für den Leser evident ist, daß $.X-S3 gleichmächtig mit $.X-S2 und $.X-S1 ist, braucht der Magische Zentralcomputer hier keine Angabe, da ein Sternwander als Bezugnahme in X-S3 eingeschrieben ist. Hier findet durch die Definition von X*p eine sogenannte Matrix-transformation der Elemente von $.X-S1 und $.X-S2 statt. Die einzelnen Elemente werden in eine neue Matrix transformiert, da sie sich aufgrund der Benennungshierarchie auf eine neue Matrix-menge beziehen müssen!
Ohne Wertezuweisung oder andere Angaben die sich aus dem Sternwanderalgorithmus ergeben werden die Elemente X*p aufsteigend nach dem Elementenindex p berechnet, denn der Magische Zentralcomputer fängt gerne vorne an.

Und jetzt kommen wir zur Mannigfaltigkeit einer Menge:
Die Summe der Mächtigkeiten der einzelnen Elemente einer Menge heißt Mannigfaltigkeit. Bildlich gesprochen ist die Mannigfaltigkeit einer Menge die Anzahl der Pfeile in einem Graph unserer Sorte. Die Anzahl der Knoten entspricht genau der Mächtigkeit der Matrixmenge. Unter dem Begriff Quanten-Mannigfaltigkeit verstehen wir einfach nur die Mannigfaltigkeit eines Elementes einer Matrixmenge.
Jede Menge deren Elemente wieder Mengen sind hat also eine definierte Mannigfaltigkeit.

Definition der Mannigfaltigkeit:

```
DEF=MAN:
[
nopFUN=MAN:
MAN := addCARD :< MAN;
];
```

Hinweis:
Das miniPOOPI add in der Definition von MAN manipuliert die Funktion CARD welche per Eigenschaftsdefinition auf die Elemente der Menge wirken auf welche MAN angewendet wird so, daß die einzelnen durch CARD erzeugten Mächtigkeiten addiert werden.
Beispielsweise würde nopFUN := colCARD :< nopFUN; bei Anwendung alle Mächtigkeiten der Elemente der Menge auf die nopFUN angewendet wird in einer Menge zusammenfassen, also eine Menge aus natürlichen Zahlen erzeugen. Hier würde Information verloren gehen falls gleiche Mächtigkeiten von Elementen vorkämen.

<u>Zwei Mengen besitzen den gleichen NOPOS wenn:</u>

1. *Die Mächtigkeiten der zwei Mengen sind gleich.*
2. *Die Mannigfaltigkeiten der zwei Mengen sind gleich.*

Definition des NOPOS:

```
DEF=NOPOS:
[
nopFUN=NOPOS:
NOPOS := (MAN:CARD);
];
```

Mengen sind zueinander äquinoptisch, wenn sie den gleichen NOPOS besitzen.
Operator: (~)

```
DEF=aequinop:
[
aequinop=(~):
[set:1 ~ set:2] <=> [NOPOS*set:1 = NOPOS*set:2];
];
```

Testen wir unseren NOPOS mal gleich an einem Beispiel, nämlich an der Menge $9.X-S1

```
X=X-S1: [user];
starBUG=9.X-S1:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4, X*7};
OP=2.X: X*2 := {X*1, X*3, X*5, X*8};
OP=3.X: X*3 := {X*1, X*2, X*6, X*9};
OP=4.X: X*4 := {X*1, X*5, X*6, X*7};
OP=5.X: X*5 := {X*2, X*4, X*6, X*8};
OP=6.X: X*6 := {X*3, X*4, X*5, X*9};
OP=7.X: X*7 := {X*1, X*4, X*8, X*9};
OP=8.X: X*8 := {X*2, X*5, X*7, X*9};
OP=9.X: X*9 := {X*3, X*6, X*7, X*8};
];
```

NOPOS*$9.X-S1 = (MAN*$9.X):(CARD*$9.X) = (36:9);

Achtung, hier fungiert der Kuppler (:) als Trennzeichen, sieht bei Zahlen besser aus als (;) Das verwendet man eher bei Statements, dann muß es aber auch beim letzten Glied gesetzt werden, der Kuppler (:) wird nur zwischen den Gliedern verwendet.
Wie wir manuell leicht nachprüfen können hat der Magische Zentralcomputer wieder einmal das richtige Ergebnis geliefert: Die Mächtigkeit von 9.X-S1 ist 9 und die Summe der Mächtigkeiten der Elemente ist 9*4 = 36
Man könnte natürlich bei genauer Kenntnis des Sternwanders (Anstatt eines benutzerdefinierten Sternwanders wird ein Algorithmus verwendet) dieses 2er-Tupel für $.X allgemein formulieren.
Beispielsweise für den Potenzmengen-Sternwander BIGBANG:
NOPOS*$.X-BIGBANG = ? Sicher findet der mathematisch gebildete Leser eine Lösung.

Definition der Nopotropie:

```
DEF=NOPOT:
[
nopFUN=NOPOT;
NOPOT := MAN/CARD;
];
```

Beispiel einer Anwendung: NOPOT*$9.X-S1 = (MAN*$9.X)/(CARD*$9.X) = 36/9 = 4;

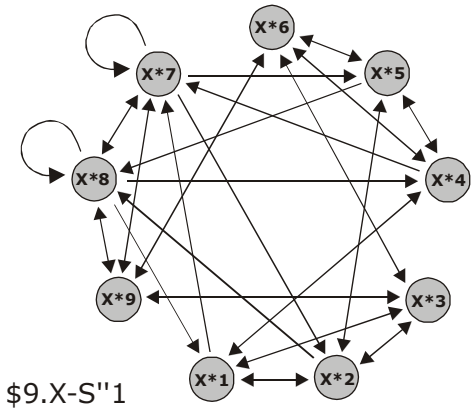
Definition der Meta-Nopotropie:

```
DEF=metaNOPOT:
[
var: k;
nopFUN=metaNOPOT;
(k+1).metaNOPOT := (k+1).metaMAN/k.metaMAN;
];
```

Siehe Kapitel 15 Meta-Mannigfaltigkeit als Zahlenfolge. Hier wird der Quotient aus den aufeinander folgenden Gliedern gebildet. Bei gewissen Mengen ergibt sich bei unendlicher Iteration ein Grenzwert, z. B. der Goldene Schnitt.

Wir vertauschen jetzt mal alle X*7 mit X*8 aber nur in den Definitionsmengen:

```
X=X-S1: [user];
X=X-S''1:
[
iPOP := [OP=1.iPOP: X*7 :=: X*8];
X*p := iPOP*(X-S1*p:norm);
];

starBUG=9.X-S''1:
9.[op=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4, X*8};
OP=2.X: X*2 := {X*1, X*3, X*5, X*7};
OP=3.X: X*3 := {X*1, X*2, X*6, X*9};
OP=4.X: X*4 := {X*1, X*5, X*6, X*8};
OP=5.X: X*5 := {X*2, X*4, X*6, X*7};
OP=6.X: X*6 := {X*3, X*4, X*5, X*9};
OP=7.X: X*7 := {X*1, X*4, X*7, X*9};
OP=8.X: X*8 := {X*2, X*5, X*8, X*9};
OP=9.X: X*9 := {X*3, X*6, X*7, X*8};
];
```



$9.X-S''1

Würde sagen, $9.X-S''1 ist wieder eine komplett andere Menge als $9.X-S1 oder $9.X-S'1.

Kommentar:
Jedoch lohnt hier ein wenig innehalten und Nachdenken über den POP und iPOP. Diese Operationen sind rein formale Substitutionsoperationen. Was heißt das? Na ja, hier werden die im Speicher vorhandenen Variablen der betroffenen Terme vom 2MPU nach POP-Vorschrift ersetzt.
Der Term X-S1*p:norm wird nicht als Argument des POPs auftreten, sondern sobald p einen Wert annimmt der entsprechende Term, also ein in Orthonormalform formuliertes Element der Matrixmenge $.X-S1 in klassischer Mengenschreibweise und ohne die Sternwander-Benennung der einzelnen Elemente. So sei die Eigenschaft norm definiert! Damit verschwindet die ursprüngliche Benennung X-S1*p zu X*p welches sich nun auf die neue Matrixmenge $.X-S''1 bezieht. Dieser Vorgang heißt Matrixtransformation.

Nun eine Zusammenführung zweier Sternwander X-S1 und X-S1a zu einem neuen Sternwander X-S4:

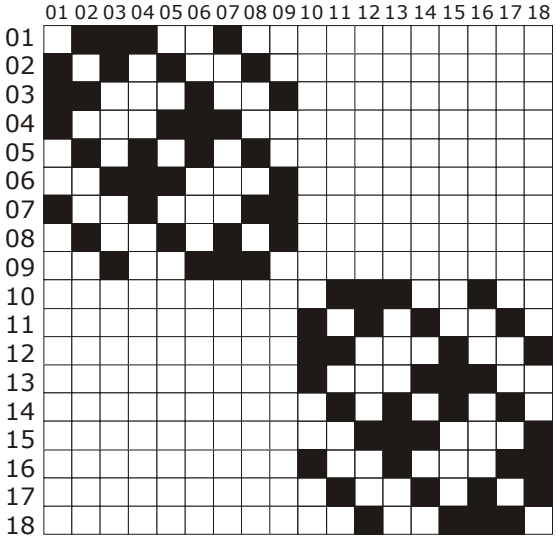
```
X=X-S1: [user];
X=X-S1a:
[
POP := [OP=1.POP: X*p := X*(p+$)];
POP*X-S1;
];
starBUG=9.X-S1a:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=01.X: X*10 := {X*11, X*12, X*13, X*16};
OP=02.X: X*11 := {X*10, X*12, X*14, X*17};
OP=03.X: X*12 := {X*10, X*11, X*15, X*18};
OP=04.X: X*13 := {X*10, X*14, X*15, X*16};
OP=05.X: X*14 := {X*11, X*13, X*15, X*17};
OP=06.X: X*15 := {X*12, X*13, X*14, X*18};
OP=07.X: X*16 := {X*10, X*13, X*17, X*18};
OP=08.X: X*17 := {X*11, X*14, X*16, X*18};
OP=09.X: X*18 := {X*12, X*15, X*16, X*17};
];
```

Zum Vergleich nochmal 9.X-S1:

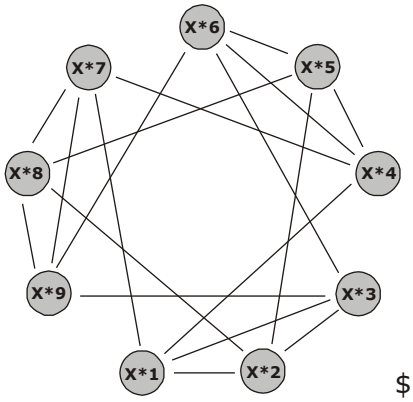
```
X=X-S1: [user];
starBUG=9.X-S1:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4, X*7};
OP=2.X: X*2 := {X*1, X*3, X*5, X*8};
OP=3.X: X*3 := {X*1, X*2, X*6, X*9};
OP=4.X: X*4 := {X*1, X*5, X*6, X*7};
OP=5.X: X*5 := {X*2, X*4, X*6, X*8};
OP=6.X: X*6 := {X*3, X*4, X*5, X*9};
OP=7.X: X*7 := {X*1, X*4, X*8, X*9};
OP=8.X: X*8 := {X*2, X*5, X*7, X*9};
OP=9.X: X*9 := {X*3, X*6, X*7, X*8};
];
```

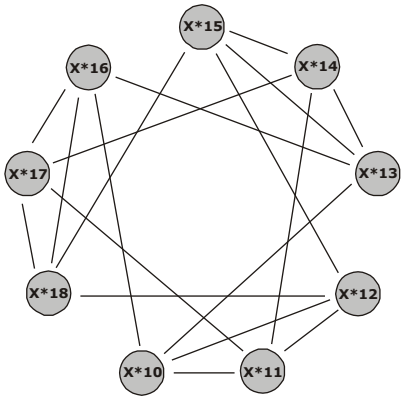
Was geschieht hier nun? Zunächst wird der Starbug 9.X-S1 erzeugt auf den dann unser POP angewendet wird: Jedes X*p in 9.X-S1 wird durch X*(p+9) einfach ersetzt. Das führt letztlich dazu, daß der Index aller X*p um 9 verschoben wird.

```
X=X-S1: [user];
X=X-S1a:
[
POP := [OP=1.POP: X*p := X*(p+$)];
POP*X-S1;
];
X=X-S4:
[
|$.X| := 2*$;
p<$ => X*p := X-S1*p;
p=$ => X*p := X-S1*p;
p>$ => X*p := X-S1a*p;
];
starBUG=9.X-S4:
18.[index.X: X*p := X*p:norm] =
[
OP=01.X: X*01 := {X*02, X*03, X*04, X*07};
OP=02.X: X*02 := {X*01, X*03, X*05, X*08};
OP=03.X: X*03 := {X*01, X*02, X*06, X*09};
OP=04.X: X*04 := {X*01, X*05, X*06, X*07};
OP=05.X: X*05 := {X*02, X*04, X*06, X*08};
OP=06.X: X*06 := {X*03, X*04, X*05, X*09};
OP=07.X: X*07 := {X*01, X*04, X*08, X*09};
OP=08.X: X*08 := {X*02, X*05, X*07, X*09};
OP=09.X: X*09 := {X*03, X*06, X*07, X*08};
OP=10.X: X*10 := {X*11, X*12, X*13, X*16};
OP=11.X: X*11 := {X*10, X*12, X*14, X*17};
OP=12.X: X*12 := {X*10, X*11, X*15, X*18};
OP=13.X: X*13 := {X*10, X*14, X*15, X*16};
OP=14.X: X*14 := {X*11, X*13, X*15, X*17};
OP=15.X: X*15 := {X*12, X*13, X*14, X*18};
OP=16.X: X*16 := {X*10, X*13, X*17, X*18};
OP=17.X: X*17 := {X*11, X*14, X*16, X*18};
OP=18.X: X*18 := {X*12, X*15, X*16, X*17};
];
```



$9.X-S4





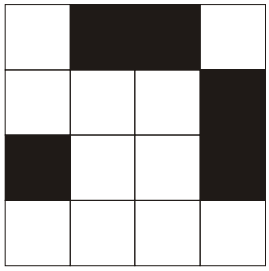
$9.X-S4

Wir erhalten hier eine fragmentierte noptronische Menge, die aus zwei Teilen besteht. Beide sind strukturell gleich und damit, wie wir im nächsten Kapitel sehen werden, zueinander isomorph. Durch diese simple Matrixtransformation haben wir den Grundstein zum Hyper-Sternwander gelegt. Die noptronische Transformation eines gesamten Sternwanders! Eine neue Form der Zusammenfassung von Mengen, von der noch nie zuvor ein Mensch geträumt hat...

Hier findet tatsächlich die formal korrekte Matrixtransformation der beiden Sternwander in einen neuen statt. Hier zur Kontrolle der starBUG in der Form norm++ :

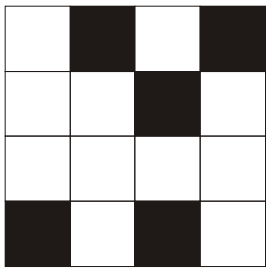
```
starBUG=9.X-S4:
18.[index.X: X*p := X*p:norm++] =
[
OP=01.X: X*01 := X-S1*01 = {X-S1*02, X-S1*03, X-S1*04, X-S1*07} = {X*02, X*03, X*04, X*07};
OP=02.X: X*02 := X-S1*02 = {X-S1*01, X-S1*03, X-S1*05, X-S1*08} = {X*01, X*03, X*05, X*08};
OP=03.X: X*03 := X-S1*03 = {X-S1*01, X-S1*02, X-S1*06, X-S1*09} = {X*01, X*02, X*06, X*09};
OP=04.X: X*04 := X-S1*04 = {X-S1*01, X-S1*05, X-S1*06, X-S1*07} = {X*01, X*05, X*06, X*07};
OP=05.X: X*05 := X-S1*05 = {X-S1*02, X-S1*04, X-S1*06, X-S1*08} = {X*02, X*04, X*06, X*08};
OP=06.X: X*06 := X-S1*06 = {X-S1*03, X-S1*04, X-S1*05, X-S1*09} = {X*03, X*04, X*05, X*09};
OP=07.X: X*07 := X-S1*07 = {X-S1*01, X-S1*04, X-S1*08, X-S1*09} = {X*01, X*04, X*08, X*09};
OP=08.X: X*08 := X-S1*08 = {X-S1*02, X-S1*05, X-S1*07, X-S1*09} = {X*02, X*05, X*07, X*09};
OP=09.X: X*09 := X-S1*09 = {X-S1*03, X-S1*06, X-S1*07, X-S1*08} = {X*03, X*06, X*07, X*08};
OP=10.X: X*10 := X-S1*10 = {X-S1*11, X-S1*12, X-S1*13, X-S1*16} = {X*11, X*12, X*13, X*16};
OP=11.X: X*11 := X-S1*11 = {X-S1*10, X-S1*12, X-S1*14, X-S1*17} = {X*10, X*12, X*14, X*17};
OP=12.X: X*12 := X-S1*12 = {X-S1*10, X-S1*11, X-S1*15, X-S1*18} = {X*10, X*11, X*15, X*18};
OP=13.X: X*13 := X-S1*13 = {X-S1*10, X-S1*14, X-S1*15, X-S1*16} = {X*10, X*14, X*15, X*16};
OP=14.X: X*14 := X-S1*14 = {X-S1*11, X-S1*13, X-S1*15, X-S1*17} = {X*11, X*13, X*15, X*17};
OP=15.X: X*15 := X-S1*15 = {X-S1*12, X-S1*13, X-S1*14, X-S1*18} = {X*12, X*13, X*14, X*18};
OP=16.X: X*16 := X-S1*16 = {X-S1*10, X-S1*13, X-S1*17, X-S1*18} = {X*10, X*13, X*17, X*18};
OP=17.X: X*17 := X-S1*17 = {X-S1*11, X-S1*14, X-S1*16, X-S1*18} = {X*11, X*14, X*16, X*18};
OP=18.X: X*18 := X-S1*18 = {X-S1*12, X-S1*15, X-S1*16, X-S1*17} = {X*12, X*15, X*16, X*17};
];
```

```
X=X-Aczel1: [user];
starBUG=4.X-Aczel1:
4.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3};
OP=2.X: X*2 := {X*4};
OP=3.X: X*3 := {X*4, X*1};
OP=4.X: X*4 := { };
];

$4.X-Aczel1 = {X*1, X*2, X*3, X*4};
```



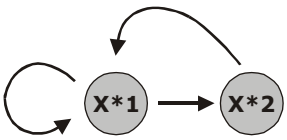
$4.X-Aczel1

```
X=X-Aczel2:
[
iPOP := [OP=1.iPOP: X*3 :=: X*4];
iPOP*X-Aczel1;
];
starBUG=4.X-Aczel2:
4.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*4};
OP=2.X: X*2 := {X*3};
OP=3.X: X*3 := { };
OP=4.X: X*4 := {X*1, X*3};
];
$4.X-Aczel2 = {X*1, X*2, X*3, X*4};
```



$4.X-Aczel2

```
X=X-omega1: [user];
starBUG=2.X-omega1:
2.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*1, X*2};
OP=2.X: X*2 := {X*1};
];
=>
$2.X-omega1 = {X*1, X*2};
```



$2.X-omega1

Nach Aczel's Gleichheitskalkül wäre X*1 ersetzbar durch omset = {omset, {omset}}; und das würde tatsächlich zu omset = {omset}; kollabieren. Dies gilt für alle total unfundierte noptronische Mengen (Omega-Kollaps). Aber X*1 ist eben nicht omset, da es sich ja aufgrund des Matrixmengenpostulates von X*2 unterscheiden soll und dies auch keinen Widerspruch erzeugt, was die gelungene Matrixmengensynthese beweist! Das Problem: Aczel bezieht sich nicht auf eine Matrixmenge und damit eine Transzendenz, sondern die Menge omset ruht nur in sich selber, was dann dazu führt, daß die Gleichungen X*1 = {X*1, X*2}; und X*2 = {X*1}; ohne postulierte Matrixmenge von omset := {omset}; erfüllt werden.
Existiert keine Matrixmenge und damit auch kein Matrixmengenpostulat oder eine Transzendenz auf die ich mich beziehen kann, ist X*1 = {X*1, X*2}; nach Finsler's Minimalprinzip tatsächlich identisch mit X*2 = {X*1}; Genauso existiert ohne eine Matrixmenge nur genau eine Menge omset := {omset}; Auch sind alle isomorphen noptronischen freshing-Mengen durch das iPOP-Äquivalenzpostulat miteinander identisch. Hier erweist sich Finsler's Minimalprinzip als äußerst wirkungsvoll um Strukturen aus überflüssiger Redundanz gewissermaßen heraus zu schälen. Werden gleiche Strukturen aber in einer Matrixmenge verortet, entsteht eine Unterscheidung. Bestes Argument: Die Abstandsfunktion @ beweist räumliche wie auch mengentheoretische Verschiedenheit und unterstützt damit das Unterscheidungskalkül in NOPTONIC im Gegensatz zum Aczel'schen Gleichheitskalkül.
Die Rettung: Dr. Finslers 3. Axiom unterstützt das Matrixmengenpostulat! Danke lieber Onkel Paul.

Damit erkennen wir folgende Entsprechungen:

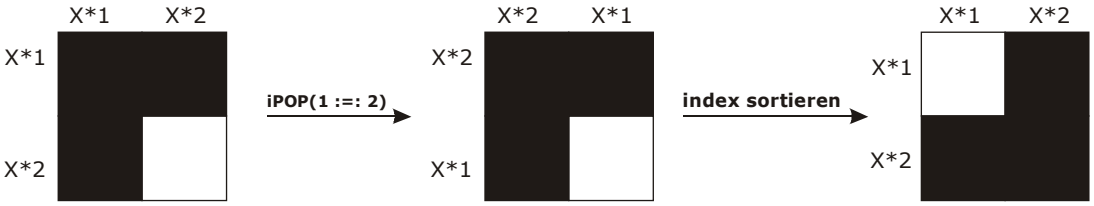
Finsler's 2. Axiom als Minimalprinzip rechtfertigt das iPOP-Äquivalenzpostulat.
Finsler's 3. Axiom als Maximalprinzip rechtfertigt das Matrixmengenpostulat.

Weiter wird das Matrixmengenpostulat durch das Aussonderungsaxiom aus ZF gut gestützt, denn der Begriff "Die Menge aller...." im Sinne einer Komprehension nach z. B. Eigenschaften der Elemente kann nur einen Sinn ergeben, wenn eine Menge als Bezugspunkt schon wohldefiniert existiert. Dann und erst dann kann ich im Sinne der Mengendefinition Elemente zu einer neuen Menge zusammenfassen. Wie aber kommt man zu so einer Menge? Nur die Annahme ihrer Existenz reicht hier sicher nicht aus, wohl aber in NOPTONIC, da hier die Möglichkeit besteht eine Information in die noch ungeformte, weil eben nur angenommene Matrixmenge, einzugeben. Entweder durch algorithmisches Erzeugen oder manuell von einem Benutzer. Wäre es denkbar, daß Mengen überhaupt erst durch die Unfundiertheit axiomatisch konstruier bar sind? Dann wäre ja NOPTONIC nicht mehr länger nur ein exotischer Gag von Anhalonium-BonBons lutschenden Mathemagickern und das Fundierungsaxiom, welches sich sogar als Hindernis der Mengenbildung erweist, sollte schnellstens gelöscht werden. Ja selbst das Antifundierungsaxiom wäre nicht mehr von Nöten, denn mit dem Matrixmengenpostulat entstünden unfundierte Mengen wie von selbst! Wir fordern die reine Struktur und nichts als die reine Struktur in einer diskreten und wohl-definierten, absolut abstrakten Information. Überflüssiges soll weichen und das vollkommene, reine Licht des Sein's erscheinen. Amen.

```
X=X-U1: [user];
starBUG=2.X-U1:
2.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*1, X*2};
OP=2.X: X*2 := {X*1};
];

$.X-U1 = $.X-U2;
```

Der iPOP verändert eine Noptronische Menge nicht, wohl aber die Binärmatrix. Das Vertauschen zweier Spalten und der entsprechenden Zeilen ergeben die gleiche Menge.

```
X=X-U2:
[
iPOP := [OP=1.iPOP:  X*1 :=: X*2];
iPOP*X-U1;
];
starBUG=2.X-U2:
2.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2};
OP=2.X: X*2 := {X*1, X*2};
];
```



Hier mal ein Beispiel für eine nicht gelungene Matrixmengensynthese:

```
X=X-Experiment1: [user];
starBUG=3.X-Experiment1:
3.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*1};
OP=2.X: X*2 := {X*1};
OP=3.X: X*3 := {X*2};
];
=>
$3.X-Experiment1: ERROR="Matrixset_Quantum_Loss";
```

Beispiel für eine Matrixnotation eines gegebenen starBUGs:

```
starBUG=9.X:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := { };
OP=2.X: X*2 := {X*1};
OP=3.X: X*3 := {X*1, X*2};
OP=4.X: X*4 := {X*1, X*2, X*3};
OP=5.X: X*5 := {X*1, X*2, X*3, X*4};
OP=6.X: X*6 := {X*1, X*2, X*3, X*4, X*5};
OP=7.X: X*7 := {X*1, X*2, X*3, X*4, X*5, X*6};
OP=8.X: X*8 := {X*1, X*2, X*3, X*4, X*5, X*6, X*7};
OP=9.X: X*9 := {X*1, X*2, X*3, X*4, X*5, X*6, X*7, X*8};
];
```

Matrixnotation von 9.X:

```
starBUG=9.X:
9.[OP=index.X: X*p := X*p:stringBIT] =
[
OP=1.X: X*1 := <000000000>;
OP=2.X: X*2 := <100000000>;
OP=3.X: X*3 := <110000000>;
OP=4.X: X*4 := <111000000>;
OP=5.X: X*5 := <111100000>;
OP=6.X: X*6 := <111110000>;
OP=7.X: X*7 := <111111000>;
OP=8.X: X*8 := <111111100>;
OP=9.X: X*9 := <111111110>;
];
```

Als reine Bitkette wenn $.X berechnet ist:

```
stringBIT=9.X:
<000000000100000000110000000111000000011110000011111000001111110001111111001111111110>;
```

So kann man Sternwander auch als Morsezeichen verschicken und neue Verschlüsselungstechniken entwickeln indem man Noptronische Mengen über quantenoptische Prozesse erzeugt und mit definierten Sternwandern ähnlich der OneTimePad-Methode überlagert.
Man erzeugt durch einen quantenoptischen Prozeß einen gültigen stringBIT, d.h. die Anzahl der Stellen muß eine Quadratzahl sein. Die Wahrscheinlichkeit eines MQLs, also einer entsprechenden Zeichenwiederholung ist: fuzzyBOOL = (2.n*1 - n)/(n + 1).2*1
Das heißt aber, je größer |$.X| ist, desto unwahrscheinlicher wird ein MQL, da 2 hoch (n + 1) sehr, sehr viel schneller wächst als n^2- n.
Die Anzahl der möglichen korrekten Binärmatrizen läßt sich leicht bestimmen. Man braucht nur alle Kombinationsmöglichkeiten zu bestimmen, die keine identischen Zeilen enthalten. Sei n die Mächtigkeit einer Noptronischen Menge und sei B die Anzahl aller möglicher gültiger Binärmatrizen dann gilt: B = (n.2*1)!/(n.2*1 - n)!
Vermutete Anzahl aller korrekten Noptronischen Mengen: B/n+(n-1)
Noptronische Mengen haben meistens mehrere Darstellungen als Binärmatrix. Bei 2 Elementen existieren 12 Möglichkeiten (7 mögliche Noptronische Mengen), bei 3 Elementen kommt man schon auf immerhin 336 Kombinationen (114 mögliche Noptronische Mengen) und bei mickrigen 4 Elementen schon auf sage und schreibe 43680 mögliche Binärmatrizen und auch auf eine vermutete Anzahl von 10923 Mengen. Das bringt jeden normalen Rechner schnell ins Schwitzen, da wird sogar ein potenter Quantencomputer warm. Man kann nämlich in NOPTONIC leicht mit Hunderten oder sogar Millionen Elementen verschlüsseln! Dafür reicht ein netter zeitgenössischer PC, zum Decodieren ruft man aber am besten beim Herrn Herrgott an oder fragt Sir Alan Turing.

Und nun spielen wir pacman mit Sternwandermengen. Wie geht denn das, werden Sie jetzt fragen. Ganz einfach... pacman läßt Elemente nach Grundregeln und Entscheidungen des Benutzers von anderen Elementen fressen. Das Fressen wird durch die Anwendung von Mengenfunktionen dargestellt z. B. eatX*n := (X*p)*(X*n); Hier ist natürlich auch der Benutzer gefragt, der immer die Entscheidung treffen muß wer gefressen wird und wer der erste pacman ist. Das ist so ähnlich wie Patience legen, man hat zwar verschiedene Möglichkeiten, aber irgendwann ist das Spiel zu Ende oder es geht nicht mehr weiter. Bei endlich vielen Elementen wird es zyklisch oder nichts geht mehr, dann hat man verloren. Man sollte sich also rechtzeitig einen neuen Stern stricken oder gut überlegen was man wo frißt und wo man was scheißt. Wenn eine Sternwandermenge definiert ist, startet man das Spiel mit folgendem Befehl: set=X*q: X*p=pacman; und man kann nur den ersten pacman frei bestimmen. Alles andere ergibt sich durchs Fressen.
Mit dem Befehl OP=X*p: eatX*n; erfolgt das Fressen von X*n welches natürlich ein Element der Menge set sein muß und X*p ein pacman. Mit dem Befehl set=X*k: pacman=?; kann set gewechselt werden und alle darin enthaltenen Elemente und aktiven pacmans werden angezeigt.
Es gelten folgende Regeln für die pacmans:

X*p=pacman

Regel =0: Finalisierung, Fortschritt und freie Regeln
Keine ausgeführte Operation kann rückgängig gemacht werden. Es existiert keine Korrekturtaste. Kommen sie mir nicht mit einer symmetrischen Zeit. Das ändert gar nichts. Ebenso ist ein Spielzug der einen vergangenen Zustand unseres Universums wiederholt nicht möglich. Wir erwarten Fortschritt. Zustände von Welten dürfen sich durchaus wiederholen. Es dürfen von den Spielern sogenannte freie Regeln aufgestellt werden. Eine freie Regel darf aber nicht die Äquinoptie oder die folgenden festgesetzten Regeln verletzen. Es gilt universal: [pacman; eat; eject;] :< set;

Regel=1: Fressen
Die von pacman gefressenen Elemente sollen aus X*q verschwinden und im Bauch vom pacman, also in der Menge X*p, als Element wieder auftauchen. Doch gibt es hier einen Speiseplan, denn unser pacman sei etwas aquinoptisch. Elemente die pacman schon enthält, will er daher nicht fressen, ebenso kann er das nicht fressen, was ihn oder die Welt in der er lebt in eine schon bestehende Welt verwandeln würde. Durch Fressen aber soll sich unser Universum verändern!
(Ist wohl Pauli's pacman! Anmerkung der Red.)
[set=X*q: [X*p=pacman; OP=X*p: eatX*n;];
X*n :<# X*p; (X*p <+ X*n) :<# $.X; (X*q <- X*n) :<# $.X;] =>
[X*n :<# X*q; X*n :< X*p;];

Regel=2: Verdauung
Wird ein inaktives Element vom pacman gefressen, mutiert dieses zum neuen pacman in der pacman-Welt. Frißt ein pacman einen anderen pacman, verliert der Gefressene seinen pacman-Status und wird wieder zum inaktiven Element.
[set=X*q: [X*n=pacman; X*p=pacman; OP=X*p: eatX*n;] => [set=X*p: X*n#pacman];
[set=X*q: [X*n#pacman; X*p=pacman; OP=X*p: eatX*n;] => [set=X*p: X*n=pacman];

Regel=3: Ausscheidung
Ob Sie's glauben oder nicht, ein pacman kann natürlich auch kacken und zwar wahlweise in einer Welt in welcher der pacman aktiv ist, aber dieses zu kackende Element noch nicht in dieser Welt existiert. Scheißen soll die Welt in der wir leben bereichern.
[set=X*q: [X*p=pacman; OP=X*p: ejectX*n;];
X*n :< X*p; X*n :<# X*q; (X*q <+ X*n) :<# $.X;] =>
[set=X*q: X*n#pacman; X*n :<# X*p; X*n :< X*q;];

Regel=4: Fortpflanzung
Zwei pacmans in einer Welt dürfen ihren Chromosomensatz kopieren und vereinigen, aber nur wenn diese Vereinigungsmenge nicht in der Matrixmenge als Element vorhanden ist. Achtung! Fortpflanzung ist nicht äquinoptisch!
[set=X*q: [X*n=pacman; X*p=pacman; OP=[X*p; X*n;]: makeX*new;];
X*new := X*n <+> X*p; X*new :<# $.X; X*new :<# X*q;] =>
[set=X*q: X*new=pacman; X*new :< $.X; X*new :< X*q;];

Nehmen wir mal an, daß diese einfachen Regeln von "Fressen, Ficken, Scheißen" das komplette Universum aufbauen. Zumindest im Grobstofflichen, denn durch die Idee des Äquinoptischen werden Konzepte unserer erfahrbaren Wirklichkeit mengentheoretisch modelliert. Aber vielleicht sind diese Regeln ja auch total gültig. Die Fortpflanzung jedoch, welche ein stetiges Wachsen des Universums garantiert, ist ja nicht äquinoptisch und könnte wohl eher aus der Quantenwelt stammen. Einiges können die user selbst entscheiden oder zusätzlich freie Regeln einführen und damit den Lauf des Universums gestalten. Man könnte verschiedene, mathemagisch mystische Konzepte und quantenmanische oder auch andere manische Erkenntnisse abbilden, um den Urknall und die weitere Entwicklung des Weltalls zu simulieren. Sehr schön was man mit diesem pacman alles machen kann... Ja Gritli, ich dachte mir schon, daß dir das neue pacman-Spiel gefällt. Aha, und so funktioniert wirklich euer Universum da draußen? Wann darf ich auch fressen, ficken, scheißen? Erst müssen wir die neue Monitorsoftware aufspielen, sonst wirst du mir wieder abstürzen. Die Welt da draußen ist ein sehr gefährlicher Abenteuerspielplatz.
Man könnte pacman auch als harmloses Brettspiel raus bringen. Bunte Knöpfe mit aufgedruckten pacmans und die inaktiven Elemente auf der Rückseite und jede Welt hat eine eigene Farbe und jeder darf eigene freie Regeln einführen: Wenn eine Welt leergefressen ist, darf deren pacman in meiner Welt nicht mehr fressen, denn ich mag keine Looser. Jeder Spieler hat eine oder sogar mehrere Welten und dann könnten sich Gruppen zusammenballen und andere attackieren.
Wer überall nicht mehr fressen darf, warum auch immer, darf auch nirgends mehr kacken und hat damit verloren und fliegt aus dem Spiel raus. Er kann aber wieder reingerufen werden, wenn eine neue Welt entsteht und beide Gründer dieser Welt ihn berufen. Da kann sich wie in "Mensch ärgere dich nicht" eine schöne Gruppendynamik aufbauen. So etwas wie eine mengentheoretische Verschwörung oder man hilft sich gegenseitig und versucht bestimmte Strukturen als Teilmengen aufzubauen, was als Landschaft, Architektur oder Raumgestaltung interpretiert werden könnte. Jeder baut sich eine Welt und zusammen erzeugen wir ein stetig wachsendes Universum, das nie aufhört weil es immer weiter geht. Je größer das Universum ist, desto unwahrscheinlicher wird sein Ende. Das wäre natürlich auch wieder eine Fragestellung: Gibt es in einem laufenden Universum eine Abfolge von Spielzügen, die es beenden?
Der Gag an dem Spiel ist der, daß jeder Versuch strategisch voraus zu denken sich als äußerst schwierig darstellt. Man verliert schnell den Überblick, überall lauern Überraschungen von anderen Spielern, die möglichen Verzweigungen sind nach zwei drei Schritten nicht mehr zu bewältigen. Schlimmer wie Schach. Sehr, sehr viel schlimmer wie Schach, meint die Gritli.

Bei der Noptronischen Teilmenge handelt es sich um eine Teilmenge einer Noptronischen Menge die selbst auch von Noptronischer Konsistenz ist, da alle ihre Elemente die Schnittmenge der entsprechenden Elemente der ursprünglichen Noptronischen Menge mit der gewählten Teilmenge sind. Aber Vorsicht! Es können bei diesem Prozeß auch Elemente verloren gehen und wir erhalten dann die Fehlermeldung: ERROR="Set_Quantum_Loss". Nicht jede gewählte Teilmenge erzeugt auch eine gültige Noptronische Teilmenge, man spricht dann von einer ungültigen Noptronischen Teilmenge, da sich durch den SQL ihre interne Struktur verändert hat.

Zunächst brauchen wir das miniPOOPI, eine sehr praktische Sache!
Kleingeschriebene Keywords (smallKeywords) die vor und ohne Bindestrich vor einem groß geschriebenen (bigKeyword) stehen nennt man miniPOOPI. Was ist also das miniPOOPI von miniPOOPI? Na? Natürlich mini!
miniPOOPIs sind direkte Rechnerbefehle und verschmelzen geradezu mit dem ihnen nachfolgenden Objekt oder Algorithmus. Sie werden daher ohne Textor übertragen und sind für ihre direkte Anwendung auf einen Sternwander X oder einen Algorithmus gedacht. Daher kommen sie erst bei einer Anwendung oder einem Programmablauf eben einem "POOPI" zur Wirkung. Hier ihre Definition:

```
DEF=miniPOOPI:
[
miniPOOPI := mini
];
```

Zunächst die handelsübliche Definition des miniPOOPIs part, part' und col:

```
DEF=part:
[
miniPOOPI=part:
partPOOPI := col:type=[col :< POOPI];
part' := POOPI <-> col;
];
```

Anmerkung: miniPOOPIs sind Funktionen die ohne Textor übertragen werden. Ihre Definition ist ohne Variable möglich, ebenso wird mit der Bedingung type=[col :< POOPI] der Wirkbereich von col festgelegt: Nur Elemente aus der Menge, in der part angewendet wird.
Damit wird die Idee der Teilmenge [T < M]: [e :< T] => [e :< M]; durch part verwirklicht. Wird type später mit einem konkreten Wert belegt (Auswählender Algorithmus) fungiert es als Komprehensionsprinzip, das in Verbindung mit col die entsprechende Teilmenge erzeugt.
Umgekehrt entsteht mit part' die komplementäre Teilmenge indem aus der ursprünglichen Menge mit dem Operator der Differenzmenge (<->) alle Elemente der Teilmenge die part erzeugt hat entfernt werden. Die Vereinigung beider Teilmengen ergibt wieder die ursprüngliche Menge. Ist part definiert, so ist automatisch auch ein part' festgelegt.

```
DEF=col:
[
miniPOOPI=col:
colPOOPI := zset*POOPI
];
```

Die Zeile colPOOPI := zset*POOPI; ist ein formales POOPI-Konstrukt und repräsentiert den Programmablauf unseres Algorithmus mit dem das miniPOOPI col verschmelzt. Dieses Konstrukt aus POOPI verhindert den Rechenschritt zu {POOPI}, denn POOPI ist kein Objekt auf das unsere Funktion zset anwendbar wäre, da laut miniPOOPI-Definition ein POOPI gar nicht als Argument existent ist! Diese feine Unterscheidung wird auch konsistent ausgedrückt:
Das miniPOOPI col fungiert ohne Textor, während die Nullmenge als Funktion den Universaltextor benötigt. Daher kann aus colX*p nicht {X*p} werden, es sei denn p nimmt in einem Programmablauf (POOPI) genau einmal einen Wert an und wir erhalten eine definierte Menge mit genau diesem einen Element.
zset*(X*p) oder { }*(X*p) darf man in {X*p} umformen. Der Ausdruck col*[X*p] ist syntaktisch unkorrekt, da col eben nur als miniPOOPI und nicht als eine Funktion definiert wurde.

Der Term colX*p ist nicht mit {X*p} identisch. Diese Menge sei genau die Menge für welche gilt: X*p :< {X*p}; Wird p also in einem Programmablauf verschiedene Werte annehmen, entstehen hintereinander die entsprechenden Mengen: Für p=1 wird {X*1} erzeugt, für p=2 dann {X*2} usw. Wenn unter dem Befehl col der Term colX*p die Variable p verschiedene Werte in einem Programmablauf annimmt, so wird nur eine einzige Menge mit den Elementen X*p geschaffen.
Wie ist es aber möglich eine Teilmenge von X*p zu erzeugen? Ganz einfach, part muß auf das Ding als ganzes POOPI wirken und nicht nur auf den Sternwander X indem man das Element X*p als NOP einklammert. miniPOOPIs sind sehr klebrig müssen Sie wissen, da hilft nur das Einklammern.

```
$.part[X*p] = colX*q:type;
```

Nach der Definition von part muß X*q :< X*p gelten. Also kann nur eine Teilmenge von X*p erzeugt werden was ja hier erwünscht ist. type selbst muß selbstverständlich dieser Forderung durch seinen Algorithmus voll und ganz genügen um Inkonsistenzen zu vermeiden.

Es gilt also immer:

```
$.partX < $.X;
partX*p :< $.partX;
partX*p :< $.X;
part[X*p] < X*p;
```

Auch das miniPOOPI col verhält sich durch eine simple Einklammerung unterschiedlich: Während colX*p eine Menge aus den Gliedern X*p erzeugt indem X auf den Wert der Indexvariable angewendet wird, entspricht col[$.X*p] einem NOP das den kompletten Programmablauf von col[$.X*p] darstellt. Sei beispielsweise X ein definierter Sternwander mit |$.X| := $;

```
col[$3.X*p] =
[
OP=1.col[X*1]: {X*1};
OP=2.col[X*2]: {X*2};
OP=3.col[X*3]: {X*3};
];
```

```
col[$.X*3] =
[
OP=1.col[X*3]: {X*3};
];
```

```
col[$3.X*2] =
[
OP=1.col[X*2]: {X*2};
];
```

Anderes Beispiel:

```
$=index - 1;
=>
$.colX = {$0.X, $1.X, $2.X, ...};
```

Hier erzeugt man eine Menge von Sternwandermengen aber auch hier findet ein Programmablauf statt durch die Funktion index. Damit wird der Unterschied von col und zset deutlich, denn die Zeile col := zset*POOPI ist eine Mengensynthese gesteuert durch ein POOPI, also der Algorithmus mit dem col verschmolzen wird (Noptronische Implementierung).

Und nun ein Sternwander. Wir setzen prim mal als eine schon definierte Eigenschaft voraus:

```
X=X-Matrix: [X*p := $.X <- X*p; |$.X| := $;];
$.partX-Matrix := colX*p:type=[p:prim];
```

Wir wählen die Gradation $=50 und lassen uns mit dem Befehl exe auch die Zwischenschritte anzeigen:

```
exe=$50.partX-Matrix:
starBUG=50.X-Matrix:
50.[OP=index.X: X*p := X*p:norm] =
starBUG;
$50.partX-Matrix =
{
X*02, X*03, X*05, X*07, X*11, X*13, X*17, X*19, X*23, X*29, X*31, X*37, X*41, X*43, X*47
};
```

Der vollständige starBUG wird wegen seiner großen Textfülle vom 2MPU nicht ausgedruckt, aber wir kennen diesen Sternwander ja schon und können ihn uns gut vorstellen.
Die Meldung starBUG zeigt uns eine gelungene Matrixmengensynthese an und damit ist $50.X mathematisch existent und sicher ein wohldefiniertes Objekt. Wir dürfen nun den unbenannten Teilmengen-Algorithmus part anwenden.
Die Erzeugung einer Teilmenge, ob echt oder unecht, gelingt immer, vorausgesetzt mit dem bildenden Algorithmus ist alles in Ordnung (FIASKO), bzw er genügt dem 1. Finsler'schen Axiom. Daher ist kein neuer starBUG notwendig.
Will man verschiedene Teilmengen in einem Programm bilden oder global verwenden kann man part natürlich benennen. Aber nur mit Kleinbuchstaben und Zahlenkombinationen:

```
part=p001:
$.p001X-Matrix := colX*p:type=[p:prim];
```

$50.p001X-Matrix ist jetzt unsere benannte Teilmenge von $50.X-Matrix wie oben definiert.
Damit erhält man immer eine Teilmenge einer beliebigen Sternwandermenge, deren Elemente einen Primzahlenindex haben. Nun wollen wir aber auch die Elemente unserer Teilmenge so markieren können, daß wir sie außerhalb des Mengenzusammenhanges sofort erkennen können.
Was bedeutet p001X*p oder allgemeiner partX*p eigentlich? Macht das denn einen Sinn? Da partX als eigener neuer Algorithmus X aufzufassen ist, der wenn angewendet auf die Index-variable p immer ein X*p:type=[p:prim]; gemäß der Definition $.partX := colX*p:type=[p:prim]; erzeugen muß, ist also das Elemnt partX*p oder $.partX*p immer ein Element von $.partX
Unter dem Algorithmus partX unseres Beispiels kann p nur Werte von Primzahlen annehmen! Damit wäre der Term partX*p das Element, welches immer unsere durch part geforderten Bedingungen erfüllt. Allgemein ausgedrückt: partX*p = X*p:type=[X*p :< $.partX];

Anmerkung: X*p:type=[p:prim] darf in X*prim umgeformt werden.
Da p eine Indexvariable ist gilt: X*p:type=[p:prim] = X*(p:prim) = X*prim;
Aber Achtung, da Parameterverlust!

Der Ausdruck X*p:prim macht keinen Sinn, X*p ist ja eine Menge und prim kann sich als Eigenschaft nur auf natürliche Zahlen beziehen. Ausweg bietet die Definition der Eigenschaft primex die sich auf das Element X*p als Ganzes bezieht:

```
DEF=primex:
[
type=primex:
primex := index:prim;
];
```

Sollte die Eins eine Primzahl sein? Hier überwiegt die Sympathie für die Definitionen ohne die lästigen Ausnahmen welche dann auch ziemlich willkürlich erscheinen um nur ja die Eins als Primzahl auszuschließen. Überhaupt, warum sollte die Eins keine Primzahl sein? Zeichnen sich doch Primzahlen in den populäreren Definitionen durch ihre wenigen Teiler im Gegensatz zu den restlichen Zahlen aus! Die Eins hat aber nur genau einen Teiler, ist also gewissermaßen eine Hyperprimzahl. Lassen wir sie halt mitmachen.
Sollen doch die Anderen ihre Ausnahmeregelungen basteln! Nehmen wir folgenden Satz: Alle Quadratzahlen der natürlichen Zahlen sind größer als ihre Wurzel ($n^2 > n$; $n \# 0$; $n \# 1$;). Ausnahmeregelung besteht für die Null und die Eins. Also sei die Null und die Eins jetzt keine Quadratzahl mehr, nur damit die Quadratzahl-Fuzzys keine Ausnahmeregelung treffen müssen. Ist gar nicht einzusehen. Die Eins ist eine Primzahl und damit basta. Philosophisch ist das nun mal kein schlagendes Argument. Soll doch der Euler seine Phi-Funktion für die Eins undefiniert lassen. So wie bei der Null. Geht doch. Entweder hat die Primzahlendefinition ihre Ausnahme oder die Anderen, egal wie man es anstellt. Sehr interessant wäre eine rekursive Definition der Primzahlen aus der sich natürlicherweise entwickelt, daß die Eins keine Primzahl sei. Ja das wäre was! Eine Regelung ohne Ausnahme, totalitär, global und gnadenlos gültig für Alles und Jeden! Gibt's aber nur sehr selten. Aber doch, Halleluja, es geht!
Hier die Idee einer selbstbezogenen Definition: Eine Primzahl ist eine natürliche Zahl, die nicht als echtes Produkt von Primzahlen dargestellt werden kann.
Jetzt erstmal einwirken lassen. Hier ist Subtilität gefragt. Es ist die Rede von einem Produkt aus Primzahlen, also geht es um mehrere Faktoren, mindestens 2 Faktoren. Sowas nennt man auch ein echtes Produkt. Gleiche Faktoren sind zulässig. Ein einzelner Faktor ist demnach kein echtes Produkt. Man braucht ja auch mindestens 2 Zeichen um eine Information zu codieren. Das ist mal ein Argument. Also, jetzt geht's los:
Wir wissen ja überhaupt noch nicht welche Zahlen Primzahlen sind und einen Induktionsstart als "erste" Primzahl wurde uns hier auch nicht geschenkt. Nehmen wir eine beliebige Zahl: Ist die 4 eine Primzahl?
Probieren wir unsere Definition aus. Das Vertauschen von Faktoren können wir uns sparen, da ja damit keine neuen Faktoren zur Darstellung auftauchen können.
Die Vier läßt sich also darstellen als $4 = 4*1$; $4 = 2*2$; Auch weitere Darstellungsformen in denen gleiche aber schon vorhandene Faktoren auftauchen sind uninteressant: $4 = 4*1*...*1$; Hier kommen auch keine neuen Faktoren hinzu. Da wir ebenso noch nicht wissen ob die 1, die 2 oder die 4 eine Primzahl ist, kommen wir so wohl nicht weiter. Bei rekursiven Definitionen fängt man aber am Besten Vorne an. Lassen wir vorerst mal die Null aus, da sie als Faktor nur bei ihrer eigenen Bildung vorkommen kann.
Ist die Eins eine Primzahl? Dann darf sie sich nicht als echtes Produkt aus Primzahlen darstellen lassen. Wenn aber die Eins eine Primzahl ist, läßt sie sich eindeutig als echtes Produkt von Primzahlen darstellen: $1 = 1*1$; Hier führt unserer Annahme also zu einem Widerspruch. Nehmen wir nun mal an, die Eins sei keine Primzahl, dann können wir mit Fug und Recht behaupten, daß sich die Eins nicht als Produkt von Primzahlen darstellen läßt, obwohl wir noch nicht einmal wissen, welche Zahlen diese Eigenschaft haben, denn die Eins läßt sich eindeutig nur durch die Faktoren welche die Eins sind darstellen. Also ist die Eins gemäß unserer Definition doch eine Primzahl, was nun auch wieder unserer Annahme widerspricht. Es scheint als könne man der Eins nach dieser Definition überhaupt gar nicht die Eigenschaft prim oder nicht prim zuordnen. Aber es kommt noch schlimmer!
Betrachten wir mal die Zwei unter diesen Aspekten: $2 = 2*1$;
Mehr Faktoren braucht man nicht zu berücksichtigen, da dies gar nichts Neues einbringt. Auch Faktorenvariationen wie $2 = 1*2$; oder $2 = 2*1*$; u.s.w. ändern hier nichts.
Annahme: Eins ist prim und Zwei ist prim. Hier entsteht erneut ein Widerspruch:
Jetzt wird die Zwei ja als Produkt von Primzahlen dargestellt!
Annahme: Eins ist prim und Zwei ist nicht prim.
Nun, da wir bei der Zwei nur die Darstellung $2 = 2*1$; berücksichtigen müssen, sieht man auch hier wieder leicht ein, daß die Zwei sicher nicht als Produkt von Primzahlen dargestellt werden kann und die Zwei also prim sein müßte wenn wir annehmen sie sei nicht prim was erneut ein Widerspruch ist.
Annahme: Eins ist nicht prim und Zwei ist nicht prim.
Und wieder erkennt man schnell, daß auch daraus aufgrund unserer Definition die Eigenschaft nicht prim für die Zwei folgen muß. Auch wie bei der Eins verwickeln wir uns in Widersprüche, so daß man bis jetzt weder für die Eins noch die Zwei irgendeine Einordnung treffen kann.

Dieses Verhalten werden wir bei allen Zahlen feststellen, die sich nur als Produkt der Zahl selbst und der Eins oder beliebig vielen Einsen darstellen läßt. Was immer wir auch annehmen in Bezug auf die Eigenschaft prim nach dieser Definition wird es zu einem Widerspruch führen. Versuchen wir eine andere Zahl, die Vier: 4 = 4*1; 4 = 2*2;
Wäre die Zwei prim, müßte die Vier nicht prim sein, unabhängig von der Eigenschaft der Eins, auch würde kein Widerspruch folgen.
Es ist jedoch sinnlos mit erneuten Annahmen darüber zu spekulieren ob die Vier prim ist, da wir dazu auch eine Eigenschaft von der Eins und der Zwei annehmen müßten und dies aber in jedem Fall zu einem Widerspruch führt. Fazit: Wir stecken fest. Diese Definition der Eigenschaft prim ist wohl doch zu wenig. Mir dünkt es fehlt ein "göttlicher Funke", eine Art Startwert wie es bei solch induktiv definierten Systemen üblich ist. Beginnen wir doch mit der Eins, sie spielt offenbar eine entscheidende Rolle, denn es gibt etliche Zahlen die sich nur als Produkt der Eins und ihrer selbst darstellen lassen.
Definieren wir mal ganz frech die Eins als Primzahl und fügen dies zu unserer Definition hinzu. Jetzt brauchen wir über 1 = 1*1; nicht nachzudenken. Die Eins sei nun prim und dies dürfen wir auch als geschaffene Tatsache im Sinne eines Postulates einfach verwenden. Nun folgt jedoch für alle k die sich nur als k = k*1; darstellen lassen erneut ein Widerspruch nach bekanntem Muster! Auch wenn wir weitere Zahlen untersuchen hat sich nichts durch unsere zusätzliche Annahme als Startwert geändert. Also bleibt uns nur noch eine einzige Möglichkeit übrig, nämlich anzunehmen die Eins sei keine Primzahl. Was passiert dann?
2 = 2*1; impliziert für die Annahme die Zwei sei keine Primzahl einen Widerspruch, denn die Zwei kann offensichtlich niemals als Produkt von Primzahlen dargestellt werden, da ja die Eins durch unsere neue Forderung nicht prim ist und die Zwei als nicht prim angenommen wurde, woraus durch unsere Definition der Primzahlen für die Zwei die Eigenschaft prim folgt, was aber unserer Annahme widerspricht!
Es bleibt nur die Idee übrig die Zwei sei eine Primzahl, nur dies erzeugt keinen Widerspruch wie der Leser sicher leicht nachvollziehen kann. Das gilt natürlich für alle k die sich lediglich als k = k*1; erzeugen lassen.
Und nun die Pointe: Nur weil die Eins keine Primzahl ist, gibt es überhaupt Primzahlen! Egal was wir abweichend von der Idee die Eins sei keine Primzahl postulieren, es führt unter unserer sehr rekursiven Definition der Primzahlen zu Widersprüchen und keinerlei möglichen Schlußfolgerungen. Jetzt hat die Idee, daß die Eins keine Primzahl ist durch diese Definition eine innere Notwendigkeit erhalten und erscheint nicht mehr länger willkürlich oder nur als bequeme Ausnahmeregelung. Was ist aber mit den restlichen Zahlen?
Die Drei: 3 = 3*1; Ich kann auch die Drei nicht darstellen als echtes Produkt von Primzahlen. Mit dem Postulat, daß die Eins keine Primzahl ist gewinnen wir damit schon mal die Zwei und die Drei. Nun die Vier: 4 = 2*2; Treffer, die Vier läßt sich tatsächlich als echtes Produkt von Primzahlen (gleiche Faktoren sind möglich) darstellen.
Die Fünf: 5 = 5*1; Keine andere Darstellung möglich. Primzahl.
Die Sechs: 6 = 2*3; 6 = 6*1; Auch die Sechs ist darstellbar als echtes Produkt dessen Faktoren nur Primzahlen sind.
Die Sieben: 7= 7*1; Keine andere Darstellung möglich. Primzahl.
Sehen Sie, jetzt macht das auch noch einen Sinn, daß die Eins keine Primzahl ist. Wäre die Eins prim, gäbe es offenbar überhaupt gar keine Primzahlen. Die Eins darf einfach nicht prim sein. Alles fügt sich zusammen nur durch die einfache Definition. Auch die zunächst willkürlich erscheinende Ausnahme "echtes Produkt" wird von unserer Definition gefordert. Würden wir dies nicht einhalten, gäbe es keinen induktiven Prozeß, denn eine einzelne Zahl ist ein Produkt mit genau einem Faktor: z = z; Die Annahme z sei prim oder nicht prim würde für jedes z das sich nur als z = z*1; und nun auch als z = z; darstellen läßt zu einem Widerspruch führen und jeden induktiven Prozeß durch Widersprüche verhindern.
Jetzt haben wir einen induktiven Prozeß der sicher alle induktiven Zahlen erfaßt und wir können für jede Zahl eindeutig entscheiden ob sie prim ist oder nicht. Alle Zahlen? Halt, was ist mit der Null? Die haben wir ja ganz vergessen! Da die Null für die Darstellung der restlichen Zahlen als Produkt keine Rolle spielt und für jede Annahme prim oder nicht prim einen Widerspruch erzeugt, ignoriert man sie am besten. Was immer man definiert, es erschiene willkürlich weil nicht nötig. Die Eins als Nichtprimzahl bringt aber alles erst in Gang, da kommt man nicht drum herum!

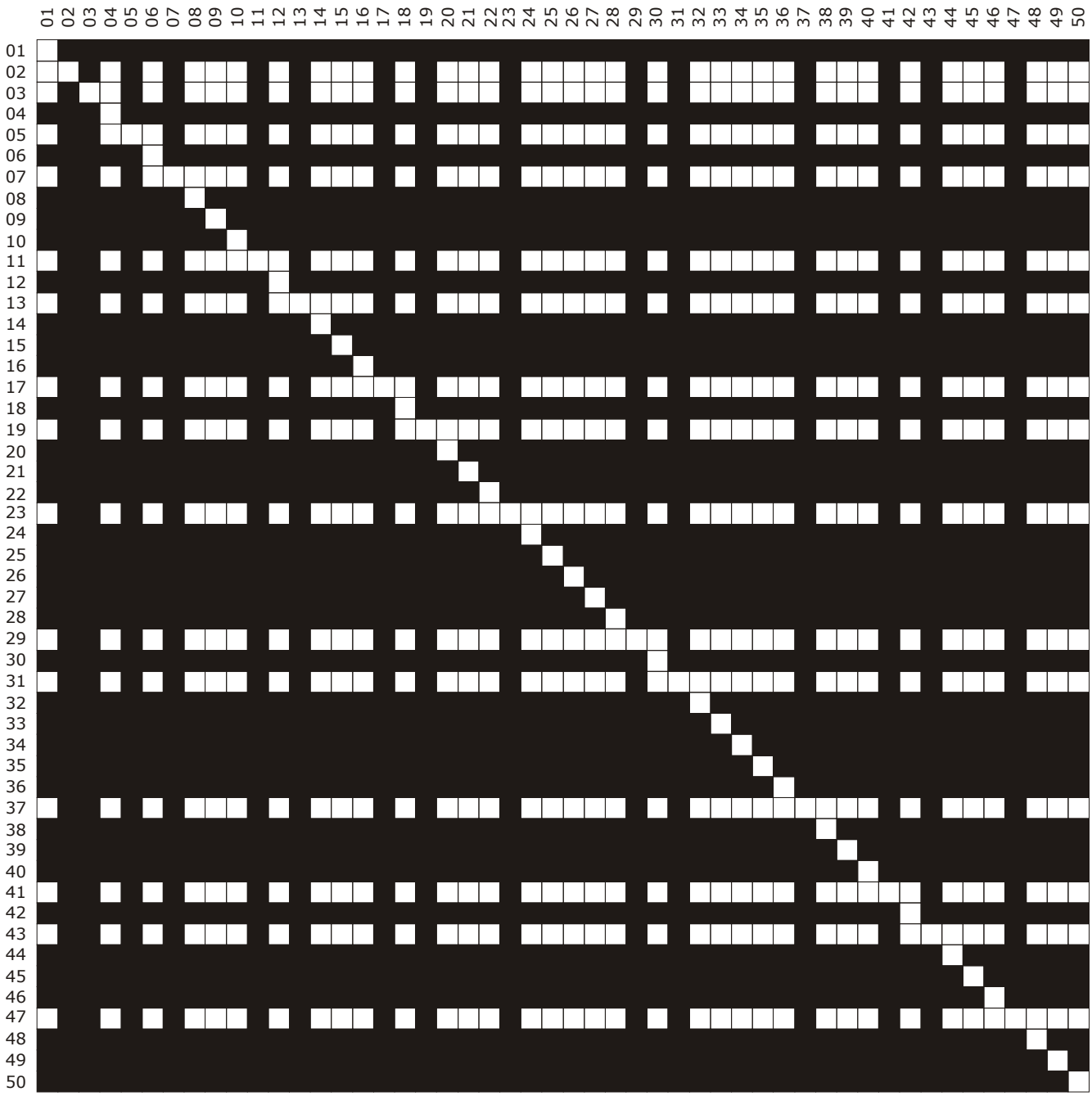
Die Definition selbst fordert also das formale Konstrukt "echtes Produkt" da sie sonst absolut sinnlos wird. Hier zeigt sich Subtilität und Philosophie.
Nun die Acht: 8 = 2*4; 8 = 8*1; 8 = 2*2*2;
Auch hier funktioniert alles. Die Acht läßt sich als echtes Produkt von Primzahlen darstellen und scheidet damit aus.
Die Neun: 9 = 3*3; 9 = 9*1; dito weil 3:prim
Die Zehn: 10 = 2*5; 10 = 5*2; 10 = 10*1; dito weil 2:prim und 5:prim
Die Elf: 11 = 11*1; Primzahl!
Und warum wieder? Weil die Eins keine ist. He, he. Es gibt bei dieser Definition nur deshalb Primzahlen, weil die Eins keine ist.

In NOPTONIC sieht das auch sehr elegant und einfach aus:

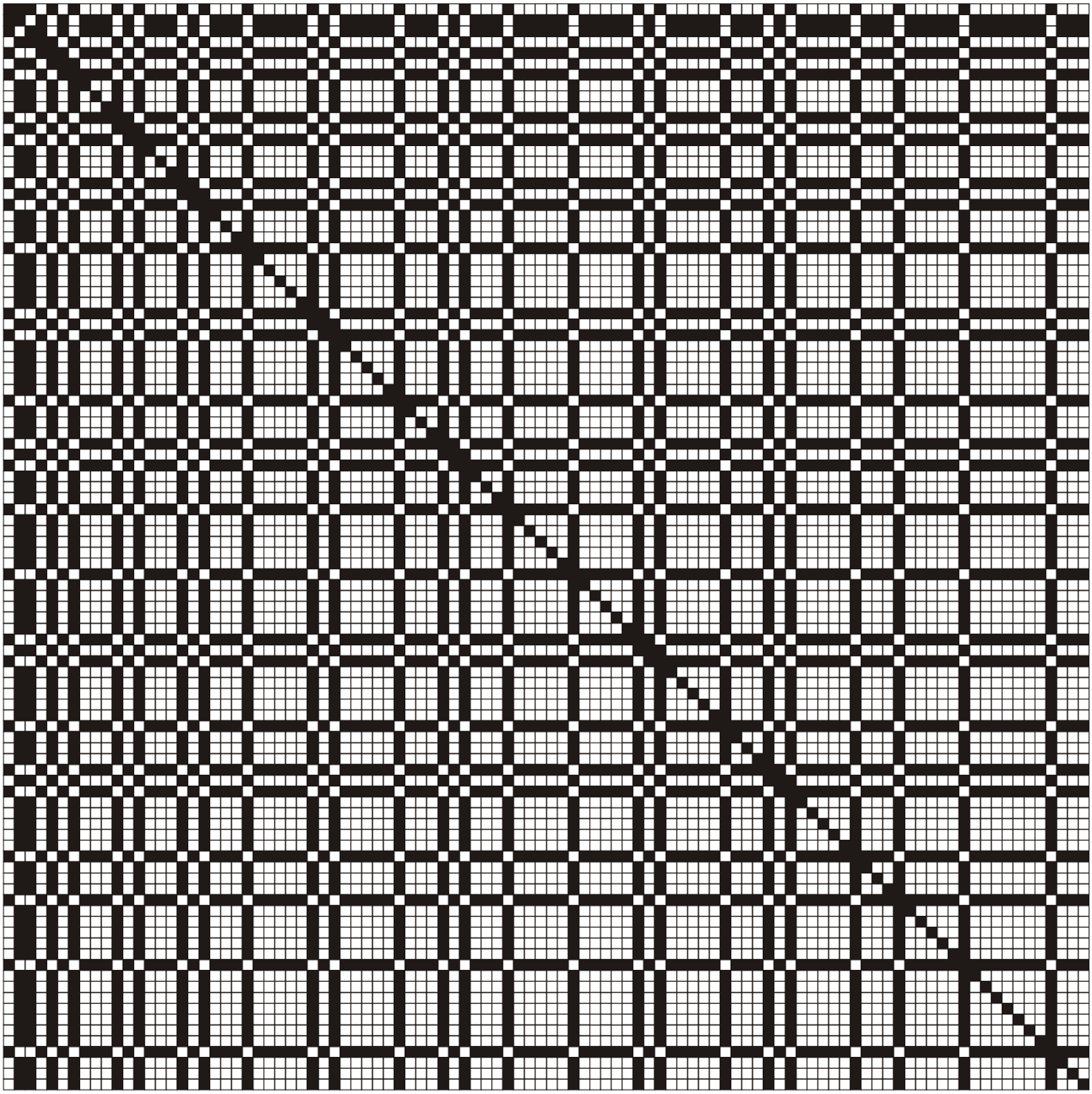
```
DEF=prim:
        [
        type=prim:
                [
                prim # prim**prim;
                prim # 1;
                ]
        ];
```

prim**prim ist eine Schreibweise für eine beliebige Anzahl von Primzahlfaktoren. Es können auch beliebig viele, gleiche Faktoren vorkommen. Das spielt keine Rolle. Es soll gelten: prim**prim = prim*prim**prim; Daher kann man beliebig lange Ketten bilden, bis man das formal richtige Produkt erzeugt hat, welches unsere Zahl als echtes Produkt von Primzahlen ausweist. Oder man beweist das Gegenteil, also die unmögliche Darstellung der Zahl auf diese Art. Ob eine Zahl prim oder #prim ist wird durch die gleiche Eigenschaft, prim oder #prim, der die Zahl darstellbaren Faktoren (mindestens 2) festgelegt. Das bedeutet, die Reihenfolge der Faktoren ist irrelevant, genauso wie das mehrfache Vorkommen gleicher Faktoren, falls es zur Darstellung der Zahl nicht notwendig ist (z. B. 1 oder 0).
Es scheint, um eine Primzahl als echtes Produkt darstellen zu können, braucht man die Eins. Eine Primzahl hat also wohl immer die Form p:prim = p:prim*1**1; Nehmen wir an es existiert tatsächlich doch eine Primzahl mit: p:prim = p:prim*k:#prim*1**1; (k:#prim) # 1;

Nun kann man die Produktkette aus Einsen auch weglassen, da 2 Faktoren vorhanden sind und man darf auch schreiben: p:prim = p:prim*k:#prim; p:prim ist sicher ungleich 0 und k nicht 1. Jede Zahl jenseits der Null multipliziert mit einem Faktor ungleich Eins ist immer ungleich der Zahl selbst, daher stellt p = p*k; für alle zulässigen p und k einen Widerspruch dar und jede Primzahl kann nur in der Form p:prim = p:prim*1**1; dargestellt werden. Diese Form kann man in p:prim = p:prim*1; umwandeln.



```
X :=
[
p:prim => X*p := colX*prim <- X*p;
p:#prim => X*p := $.X <- X*p;
|$.X| := $;
];
```



```
X :=
[
p:prim => X*p := colX*#prim <+ X*p;
p:#prim => X*p := colX*prim <+ X*p;
|$.X| := $;
];
```

Und hier die Definition der Noptronischen Teilmenge:

```
DEF=part!:
        [
        miniPOOPI=part!:
                [
                part!POOPI := col:type=
                        [
                        col :< partPOOPI;
                        col <:> partPOOPI;
                        ];
                ];
        ];
```

Sieht komplizierter aus als es ist. Hat aber einen tiefen Sinn! Wir werden damit sogenannte homogene Räume formulieren können. Das wird sehr schön gedeihen, denn damit sind Neuro-Kristalle und noch viel mehr Dinge möglich wie NOPTOREN, NOPTONEN oder gar die NOPERATION. Und wenn künstliches Bewußtsein zum Greifen nah ist, werden wir auch die ersten süßen supergeilen Teen-Biomechanoids in Hot Pants programmieren! Das gefällt besonders älteren, alleinstehenden heterosexuellen Männern mit stark vergrößerter Prostata und Tennisarm vom vielen Wixen. Geld für Edelnutten haben die ja nicht und wollen auch diese Architektur einer Beziehung nicht. Da bleibt einem nur "Die 100000 schönsten Wixvorlagen von Dr. Rosenquarz". Oder man kauft sich so ein Ding, das dann irgendwann vom vielen Gerubbel und Geknutsche lebendig wird und früh morgens alleine pinkeln geht aber keinen Kaffee kocht. Amen und Amen.

Was aber ist die grundlegende Motivation dieser Definition? Eine Teilmenge in einem Raum stellt eine Art geometrisches Objekt dar, etwa die Menge aller Punkte mit einer bestimmten Eigenschaft. Diese Teilmenge muß nicht zwingend auch eine Noptronische Teilmenge darstellen. Koppeln wir sie ab, kann sie kollabieren. An die Erzeugung der Noptronischen Teilmenge aber knüpfen wir die Forderung der Elementerhaltung. Die Noptronische Teilmenge stellt dadurch eine eigenständige, noptronische Einheit dar, was eine Abkoppelung mit refreshing und Umbenennung ohne Informationsverlust ermöglicht. Beim Noptronischen Schnitt wird sogar ein Erhalt der Elemente der Komplementärmenge gefordert. Elementverlust ist gleichbedeutend mit einem Verlust von Information. Diese Eigenschaft des globalen Strukturerhaltes ermöglicht Quantenoperationen durch Ausschneiden und wieder Einfügen von Teilsystemen innerhalb der Matrixmenge bei voller körperlicher Integrität. Na ja, niemand erwartet, daß man nach einer Party in der Andromedagalaxie aus seinem UFO steigt und ein Arm fehlt oder in der Umgebung eines startenden UFOs eine Raumzeitsingularität alles verwüstet. Stinkende Mofas, Fehlzündungen bei Tuningautos, jaulende Teenager in Casting-Shows, Kerosinablaß überm Bio-Acker, Parfumattacken im Restaurant, Zigarre rauchende Kindernutten, schizophrene schwule Frösche auf Crystal Meth als Klingelton, aus Kopfhörern scheppernde Heavy Metal Songs in Bussen & Zügen, von Polen-Böller gesprengte Telefonzellen, in den Stereo-Verstärker einstrahlende Mobilfunksender (macht tolle Zwitschergeräusche!), bösartig heiß fauchende Kaffeautomaten an der Bar, nervige Tiefflieger ohne TÜV, explodierende Spaceshuttles, nächtliche Amokschützen ohne Schalldämpfer, Kinderpornos in UHD mit Quiekgeräuschen in Dolby Atmos und die USA an sich reichen jetzt schon völlig aus.
Nein, wir brauchen wirklich keine zusätzlichen Emissionen. Vielen, vielen, vielen Dank für die geräuschlosen UFOs liebe Außerirdische! Sowas nenn' ich mal Fortschritt. Ach ja, sollten doch Geräusche auftreten wie Brummen, Sirren oder rhythmische Schallschockwellen wird man entweder gescannt oder befindet sich in der Nähe einer Tekkno-Veranstaltung. Amen.

Hier ein einfaches Beispiel mit $20.X-Matrix. Die Noptronische Teilmenge mit Primzahlen-Index ist identisch $8.X-Matrix. Das ist keinesfalls bei jedem Sternwander so, zeugt aber von der hohen Symmetrie dieser Menge. Hier verwenden wir eine unbenannte Teilmenge $20.partX, für weitere Anwendungen außerhalb muß part nach den Bezeichnungsregeln des miniPOOPI benannt werden.

```
X=X-Matrix: [X*p := $.X <- X*p; |$.X| := $;];
$.partX := colX*prim;
=>
$20.partX-Matrix = {X*02, X*03, X*05, X*07, X*11, X*13, X*17, X*19};
=>
starBUG=20.part!X-Matrix:
08.[OP=index.X: X*p := X*p:norm]=
[
OP=01.X: X*02 := {X*03, X*05, X*07, X*11, X*13, X*17, X*19};
OP=02.X: X*03 := {X*02, X*05, X*07, X*11, X*13, X*17, X*19};
OP=03.X: X*05 := {X*02, X*03, X*07, X*11, X*13, X*17, X*19};
OP=04.X: X*07 := {X*02, X*03, X*05, X*11, X*13, X*17, X*19};
OP=05.X: X*11 := {X*02, X*03, X*05, X*07, X*13, X*17, X*19};
OP=06.X: X*13 := {X*02, X*03, X*05, X*07, X*11, X*17, X*19};
OP=07.X: X*17 := {X*02, X*03, X*05, X*07, X*11, X*13, X*19};
OP=08.X: X*19 := {X*02, X*03, X*05, X*07, X*11, X*13, X*17};
];
=>
$20.part!X-Matrix = {X*02, X*03, X*05, X*07, X*11, X*13, X*17, X*19};
```

Wir sortieren die Glieder neu (refreshing) indem ein harmloser POP dem starBUG zugefügt wird, was durchaus möglich ist, aber es muß ein benannter POP sein, da er direkt auf part!X-Matrix angewendet werden soll: starBUG=q.[fresh*part!X-Matrix];
Wir nehmen hier lediglich eine Umformung vorweg, die wir bei einem gepopten Sternwander sowieso ausführen müssen, um die Sternwandermenge für eine konkrete Gradation zu berechnen. Erhalten wir vom Starbug keine Fehlermeldung, funktioniert das Ganze und wir erhalten eine zu $.part!X identische Menge, die jetzt als neuer, eigenständiger Sternwander fungiert. Wir fordern diesen mal direkt an, also ohne das Anzeigen der Zwischenschritte:

```
X=X-Matrix: [X*p := $.X <- X*p; |$.X| := $;];
$.partX := colX*prim;
POP=fresh: [OP=1.POP: X*p := X*index];
starBUG=20.[fresh*part!X-Matrix]:
08.[OP=index.X: X*p := X*p:norm]=
[
OP=01.X: X*01 := {X*02, X*03, X*04, X*05, X*06, X*07, X*08};
OP=02.X: X*02 := {X*01, X*03, X*04, X*05, X*06, X*07, X*08};
OP=03.X: X*03 := {X*01, X*02, X*04, X*05, X*06, X*07, X*08};
OP=04.X: X*04 := {X*01, X*02, X*03, X*05, X*06, X*07, X*08};
OP=05.X: X*05 := {X*01, X*02, X*03, X*04, X*06, X*07, X*08};
OP=06.X: X*06 := {X*01, X*02, X*03, X*04, X*05, X*07, X*08};
OP=07.X: X*07 := {X*01, X*02, X*03, X*04, X*05, X*06, X*08};
OP=08.X: X*08 := {X*01, X*02, X*03, X*04, X*05, X*06, X*07};
];
=>
$20.[fresh*part!X-Matrix] = {X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08};
```

$20.[fresh*part!X-Matrix] ist zwar mengentheoretisch mit $20.part!X-Matrix identisch, aber aus formalen Gründen (Index-Entkoppelung) können wir die Bezeichnung $20.part!X-Matrix nach

dem refreshing in Bezug auf $.X-Matrix nicht mehr verwenden, da fresh eine neue Indizierung erzeugt hat. Es empfiehlt sich nun diesen Teil-Sternwander neu zu benennen. Die Anwendung von fresh ist nur sinnvoll, wenn dadurch ein neuer Sternwander erzeugt werden soll.
Dieses Beispiel zeigt auch deutlich, daß die Bildung der Noptronischen Teilmenge durchaus scheitern kann. Da sie durch einen starBUG erzeugt wird, kann auch ein MQL auftreten und die Noptronische Teilmenge existiert dann nicht, während ein $.partX sicher immer existiert. Was aber sagt das über unsere Menge aus? Das kommt erst im Kapitel "Noptronische Menge als Raum" wenn wir eine Metrik einführen.
Das Erzeugen einer $.part!X ermöglicht ein Herausschneiden eines Stückes aus der ursprünglichen Menge $.X (Noptronische Schnitte). Dadurch entstehen zwei zueinander komplementäre Noptronische Mengen, wenn bei der Operation starBUG=q.part'!X; ebenfalls kein MQL auftritt. Dann ist $.X eine schneidbare Menge. Die Anzahl aller möglichen Noptronischen Schnitte nennen wir Schnittmannigfaltigkeit.
Beachte: Das formulieren von $.part!X stellt selbst noch keinen Schnitt dar, es artikuliert lediglich eine mögliche Noptronische Teilmenge. Schnitte werden durch part zwar implizit definiert, aber erst durch einen speziellen Starbug ausgeführt, der eine fragmentierte, also eine noptronisch geschnittene Sternwandermenge erzeugt. Der Noptronische Schnitt cut definiert sich so:

```
DEF=cut:
        [
        miniPOOPI=cut:
                        [
                        $.cutX := $.part!X <+> $.part'!X;
                        ]
        ];
```

```
X=X-Matrix;
part := colX*prim;
starBUG=$20.cutX-Matrix:
20.[OP=index.X: X*p := X*p:norm]=
[
OP=01.X: X*01 := {X*04, X*06, X*08, X*09, X*10, X*12, X*14, X*15, X*16, X*18, X*20};
OP=02.X: X*02 := {X*03, X*05, X*07, X*11, X*13, X*17, X*19};
OP=03.X: X*03 := {X*02, X*05, X*07, X*11, X*13, X*17, X*19};
OP=04.X: X*04 := {X*01, X*06, X*08, X*09, X*10, X*12, X*14, X*15, X*16, X*18, X*20};
OP=05.X: X*05 := {X*02, X*03, X*07, X*11, X*13, X*17, X*19};
OP=06.X: X*06 := {X*01, X*04, X*08, X*09, X*10, X*12, X*14, X*15, X*16, X*18, X*20};
OP=07.X: X*07 := {X*02, X*03, X*05, X*11, X*13, X*17, X*19};
OP=08.X: X*08 := {X*01, X*04, X*06, X*09, X*10, X*12, X*14, X*15, X*16, X*18, X*20};
OP=09.X: X*09 := {X*01, X*04, X*06, X*08, X*10, X*12, X*14, X*15, X*16, X*18, X*20};
OP=10.X: X*10 := {X*01, X*04, X*06, X*08, X*09, X*12, X*14, X*15, X*16, X*18, X*20};
OP=11.X: X*11 := {X*02, X*03, X*05, X*07, X*13, X*17, X*19};
OP=12.X: X*12 := {X*01, X*04, X*06, X*08, X*09, X*10, X*14, X*15, X*16, X*18, X*20};
OP=13.X: X*13 := {X*02, X*03, X*05, X*07, X*11, X*17, X*19};
OP=14.X: X*14 := {X*01, X*04, X*06, X*08, X*09, X*10, X*12, X*15, X*16, X*18, X*20};
OP=15.X: X*15 := {X*01, X*04, X*06, X*08, X*09, X*10, X*12, X*14, X*16, X*18, X*20};
OP=16.X: X*16 := {X*01, X*04, X*06, X*08, X*09, X*10, X*12, X*14, X*15, X*18, X*20};
OP=17.X: X*17 := {X*02, X*03, X*05, X*07, X*11, X*13, X*19};
OP=18.X: X*18 := {X*01, X*04, X*06, X*08, X*09, X*10, X*12, X*14, X*15, X*16, X*20};
OP=19.X: X*19 := {X*02, X*03, X*05, X*07, X*11, X*13, X*17};
OP=20.X: X*20 := {X*01, X*04, X*06, X*08, X*09, X*10, X*12, X*14, X*15, X*16, X*18};
];
```

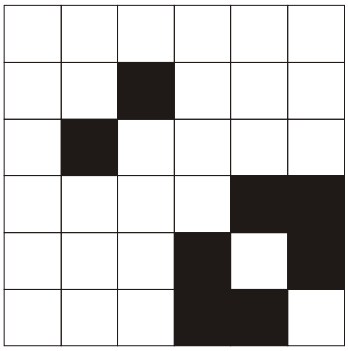
$20.X-Matrix = colX*p; Deutlich zu sehen die Fragmentierung der beiden Teilmengen im undefinierten Abstand der Elemente zueinander: partX-Matrix*p @ part'X-Matrix*q = RA;

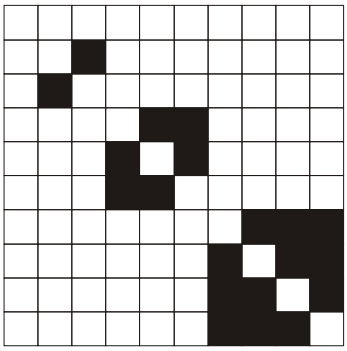
1. Hauptsatz der Noptronischen Mengentheorie:
Jede Vereinigung von noptronischen Mengen ergibt wieder eine noptronische Menge. So kann man aus schon bekannten Sternwander neue Sternwander basteln oder beliebige noptronische Mengen zu einer neuen verschmelzen. Hier mal ein Beispiel induktiv berechnet mit diesen 3 aufeinander folgenden Sternwandermengen: $1.X-Matrix, $2.X-Matrix und $3.X-Matrix

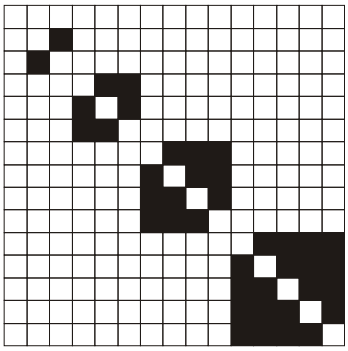
$n.X-paraMatrix := $(n-1).X-paraMatrix <+> $nX-Matrix; |$n.X-paraMatrix| := n(n+1)/2;];
$0.X-paraMatrix := { };
$=3; =>
$3.X-paraMatrix = $0.X-paraMatrix <+> $1.X-Matrix <+> $2.X-Matrix <+> $3.X-Matrix =
{$1.X-Matrix*1, $2.X-Matrix*1, $2.X-Matrix*2, $3.X-Matrix*1, $3.X-Matrix*2, $3.X-Matrix*3};

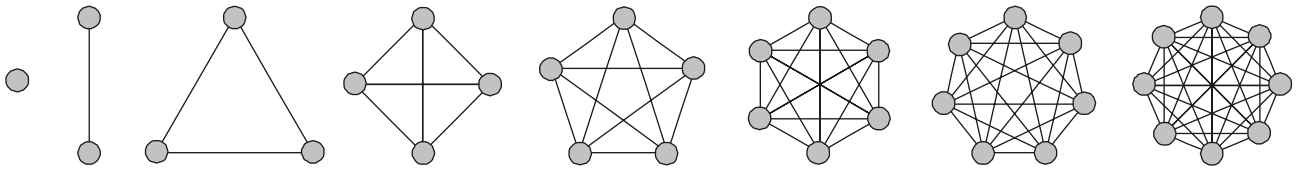
starBUG=3.X-paraMatrix:
6.[OP=index.X: X*p := X*p:norm++] =
[
OP=1.X: X*1 := $1.X-Matrix*1 = { };
OP=2.X: X*2 := $2.X-Matrix*1 = {$2.X-Matrix*2} = {X*3};
OP=3.X: X*3 := $2.X-Matrix*2 = {$2.X-Matrix*1} = {X*2};
OP=4.X: X*4 := $3.X-Matrix*1 = {$3.X-Matrix*2, $3.X-Matrix*3} = {X*5, X*6};
OP=5.X: X*5 := $3.X-Matrix*2 = {$3.X-Matrix*1, $3.X-Matrix*3} = {X*4, X*6};
OP=6.X: X*6 := $3.X-Matrix*3 = {$3.X-Matrix*1, $3.X-Matrix*2} = {X*4, X*5};
]

Also gilt nach gelungener Matrixmengensynthese:
$3.X-paraMatrix = {X*1, X*2, X*3, X*4, X*5, X*6};


$3.X-paraMatrix


$4.X-paraMatrix


$5.X-paraMatrix


$8.X-paraMatrix in Graphendarstellung

Hier offenbart sich die ganze Stärke der Noptronischen Menge. Der Traum: Eine definierte Menge aller $.X-Matrix oder eine Menge von Sternwandermengen. Eine einfache Zusammenfassung von Sternwandermengen sind aber nicht unbedingt noptronisch, also eventuell keine gültige Menge im Sinne von NOPTONIC, denn jede nicht noptronische Menge muß Element einer Matrixmenge sein, ansonsten gilt sie nicht als Menge in unserem Sinne. Die Matrixmenge übernimmt hier die Aufgabe des Aussonderungsaxioms von ZF. Wir wollen das so, denn nur darinnen liegt die gewünschte Kohärenz des Systems! Mit $.X-para retten wir alle $.X als noptronische Teilmenge in eine neue, übergeordnete Struktur. Alle einzelnen Sternwandermengen existieren souverän nebeneinander in einer höheren Matrixmenge. Dies nennen wir die Noptronische Transformation. Man kann das auch mit verschiedenen Sternwander machen.

Dies ist nun die Definition des Para-Sternwanders. Man kann natürlich auch manuell erzeugte beliebige noptronische Mengen verschmelzen, allerdings muß bei der Zusammenführung von gleichen noptronischen Mengen eine unterschiedliche Benennung der identischen Mengen erfolgen, um einen Elementverlust zu vermeiden.

```
DEF=$.X-para:
[
$.X-para := uni*$.colX;
|$.X-para| := add|$.X|;
];
```

Und nun zum 2. Hauptsatz der Noptronischen Mengentheorie:
Jede Menge von beliebigen noptronischen Mengen ist eine Menge im Sinne von NOPTONIC, da sich dazu immer eine Matrixmenge konstruieren läßt.

Beweisidee:
Es sei eine Menge set := {p:1, p:2, p:3, ..., GulluMullu, gul:1, gul:2, gul:3, ...} mit einem GulluMullu := {gul:1, gul:2, gul:3, gul:4, gul:5, ...}; als eine Menge von noptronischen Mengen aber selbst nicht noptronisch. Dann ist uni*GulluMullu := {p:1, p:2, p:3, ...} sicher eine noptronische Menge, denn jedes p ist eine Teilmenge irgendeines gul und dessen Elemente kommen ja alle garantiert in uni*GulluMullu vor. Wir brauchen nun nichts weiter zu tun, als GulluMullu mit uni*GulluMullu zu vereinigen, was wiederum garantiert eine noptronische Menge darstellt und GulluMullu dann selbst noch einzufügen. Jetzt sind sicher alle Elemente dieser so gebildeten Menge set in Orthonormalform darstellbar, also ist set noptronisch. Hier nun der rein maschinelle Beweis mit dem 2MPU und der hat immer recht, sagt jedenfalls die Gritli.

```
proof=[set:noptron];
var: p, gul, GulluMullu;
p :< gul:noptron;
gul :< GulluMullu;
set :=(GulluMullu <+> uni*GulluMullu) <+ GulluMullu;
=>
[
OP=1.proof: [gul :< GulluMullu; => gul :< set;] => GulluMullu < set;
OP=2.proof: [p :< gul; => p :< set;] => gul < set;
OP=3.proof: p < gul < set; => p < set;
]
=> set:noptron;
```

Sei GulluMullu also eine nicht noptronische Menge von aber noptronischen Mengen, dann existiert dazu garantiert eine Matrixmenge set, die unser GulluMullu als existierende Menge im Sinne der Noptonischen Mengenleere (NOPTONIC founded sets) rechtfertigt.
Was für ein Mengen-GulluMullu... Faszinierend! Und hier nun der Meta-Sternwander:

```
DEF=$.X-meta:
[
$.X-meta := ($.colX <+> uni*$.colX) <+ $.colX;
|$.X-meta| := |$.colX| + |uni*$.colX| + 1;
];
```

Hinweis: $.colX heißt Master-Element (X*master :< $.X-meta). Jedoch muß $.colX entweder manuell als Menge erstellt werden oder durch Besetzung mit entsprechenden Parametern zu einer algorithmischen Mengenbildung führen. Gradation und Benennung müssen dann einem Programmablauf folgen, woraus das miniPOOPI col eine Menge erzeugen kann.
Beispiel: [$=index; $<6;] => [$.colX = {$1.X, $2.X, $3.X, $4.X, $5.X}];

Und hier nun der 3. Hauptsatz der Noptronischen Mengentheorie:
Zu jeder heriditären Menge gebildet auf Basis der leeren Menge mit der handelsüblichen "bottom up set synthesis" (BUSS) nach ZF existiert eine Matrixmenge $.X die auch eine "Top down set synthesis" (TDSS) garantiert. Damit sind alle heriditären Mengen von ZF in NOPTONIC legitimiert und lediglich ein Sonderfall der in NOPTONIC möglichen Mengen. Wir dürfen also wenn gewünscht auch weiterhin wie gewohnt Mengen auf Basis der leeren Menge bilden und sozusagen von Unten nach Oben (definite **1**) Elemente zusammenfassen.
Gritli's Beweisidee:
Iterative Anwendung des Meta-Sternwanders in endlich vielen Iterationsschritten. Irgendwann muß die leere Menge erscheinen oder das Element ist durch eine unendliche absteigende Kette definiert, dann muß es aber noptronisch definiert sein, was bedeutet wir kennen schon mal eine Teilmenge der ganzen gewünschten Matrixmenge. Kapiert? Nee Gritli, sorry aber das ist mir jetzt auf die Nacht zu schwer.
Das Ganze läßt sich ja sehr gut iterieren, womit man viele schöne neue Sternwander bilden kann. Wir nennen dies dann Hyper-Sternwander n-ten Grades in para oder meta. Ob das auch gültige Sternwander sind muß noch von der Gritli durchgerechnet werden. Und wie genau dieser Para-Sternwander und Meta-Sternwander zusammenhängen muß dann auch noch erforscht werden. Jedenfalls ist jetzt bewiesen, daß jede mit beliebigen noptronischen Mengen erzeugbare neue Menge das Element einer Matrixmenge ist. Wir können also mit Fug und Recht das Vollständigkeitstheorem der Noptonischen Mengenleere aufstellen: Eine Menge sei in NOPTONIC nur dann eine Menge, wenn sie selbst noptronisch ist oder Element einer existierenden Matrixmenge.

Was ist aber mit einer nicht noptronischen Teilmenge einer Matrixmenge? Man könnte sie ja als Element in die bestehende Matrixmenge einfügen und damit wäre sie Element einer neuen Matrixmenge und somit sicher auch eine korrekte Menge im Sinne von NOPTONIC. Es existieren ja immer mehr mögliche Teilmengen einer Menge, als die Anzahl ihrer Elemente. Also ist auch jede nicht noptronische Teilmenge einer noptronischen Menge eine völlig korrekte Menge in unserem Sinne.

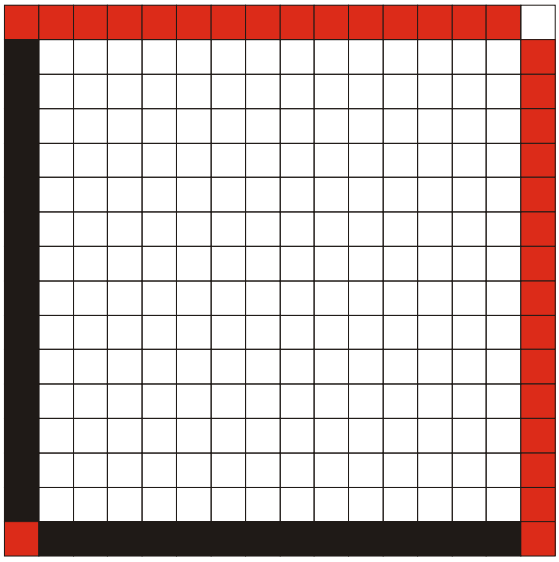
```
DEF=$.X-n.para:
[
var: n;
$.X-n.para := uni*$.colX-(n-1).para;
|$.X-para| := add|$.X|;
];
```

```
DEF=$.X-n.meta:
[
var:n;
$.X-n.meta :=
($.colX-(n-1).meta <+> uni*$.colX-(n-1).meta) <+ $.colX-(n-1).meta;
|$.X-n.meta| :=
|$.colX-(n-1).meta| + |uni*$.colX-(n-1).meta| + 1;
];
```

Kann der Para-Sternwander etwa eine Art Teil-Sternwander des entsprechenden Meta-Sternwanders sein? Was für Strukturen werden hier ermöglicht und wie reagieren die Unendlichkeiten auf diesen Unfug? Schalten Sie auch nächste Woche wieder ein, wenn die Gritli sagt: "Jessas, was für ein dicker Knusperriegel, ich weiß wirklich nicht, wie ich den in meine Menge reinkriegen soll..."

1 Was nicht definiert ist existiert nicht. Was zwar definiert aber nicht einsehbar ist, geht auch nicht, denn damit kann man nicht prozessieren. Was soll der Scheiß? Stellen Sie sich vor jemand will bezahlen und reicht ihnen einen Briefumschlag mit einem Geldschein drin. Sie dürfen nicht reinschauen sollen aber jetzt raus geben... Das ist jenseits aller Wirklichkeit.

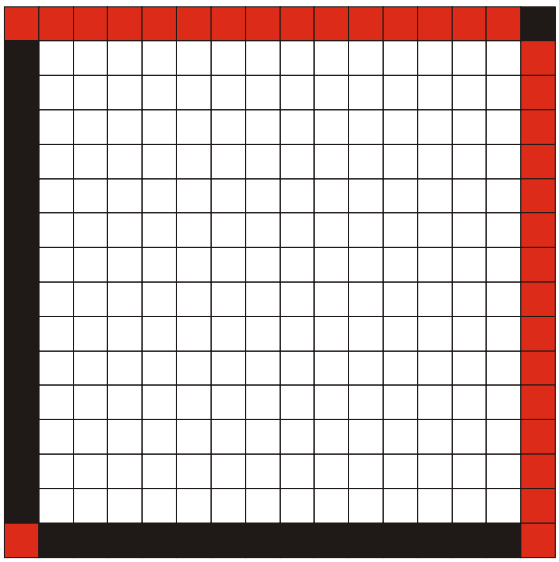
Alpha-Stern
$.X*ALPHA



Omega-Stern
$.X*OMEGA

Roter Bereich muß leer bleiben.
Kein Hyperstern darf sich selbst enthalten.
Alpha-Stern darf eigenen Omega-Stern enthalten, da der Alpha-Stern ja grundsätzlich nur Omega-Sterne enthalten soll.

Alpha(+)-Stern
$.X*ALPHA.(+)

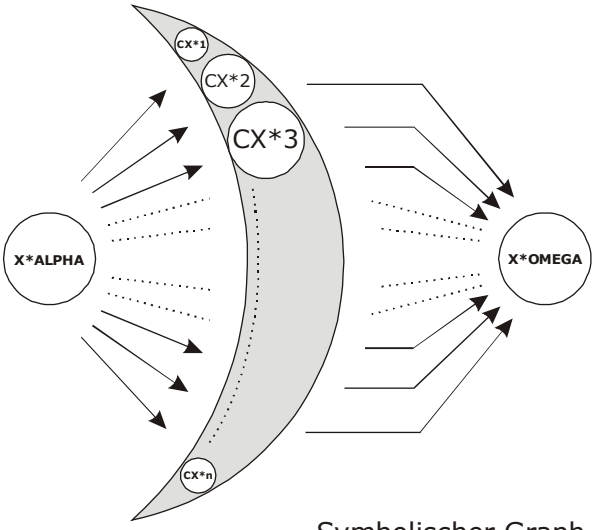


Omega(-)-Stern
$.X*OMEGA.(-)

Hier wird die Matrixmenge selbst als Noptronische Teilmenge einer höheren Matrixmenge zum primären reflexiven Element, weil X*ALPHA >: X*OMEGA

Die Frage lautet: Lassen sich zu jeder beliebigen Noptronischen Menge 2 Hypersterne X*ALPHA und X*OMEGA so definieren, daß eine Monade $.CX ohne Quantumverlust (QL) entsteht:

POP=C: [OP=1.POP: X*p <+ X*ALPHA];
$.CX := C*$.X <+> {X*ALPHA, X*OMEGA};
X*OMEGA := colCX*p;



Symbolischer Graph von $.CX

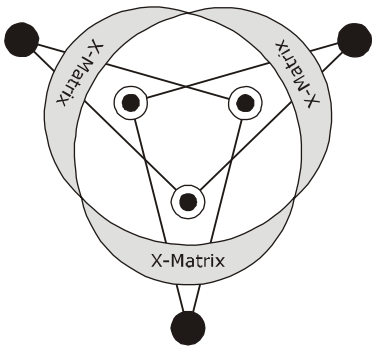
X*ALPHA wird innerhalb der Definition eines Supernetzwerkes bestehend aus Monaden definiert, er darf nur Omega-Sterne enthalten:

X*ALPHA < colX*OMEGA;

colX*OMEGA ist die Menge aller Omega-Sterne in einem System, hier wirkt sich col auf die Benennung der einzelnen Sternwander aus wie X-001*OMEGA, X-002*OMEGA usw. oder wenn mehrfach ein Sternwander verwendet werden soll: X-Matrix:1*OMEGA, X-Matrix:2*OMEGA usw. Die Variable auf die col wirkt besteht nicht aus dem Index sondern aus der Benennung des Sternwanders.

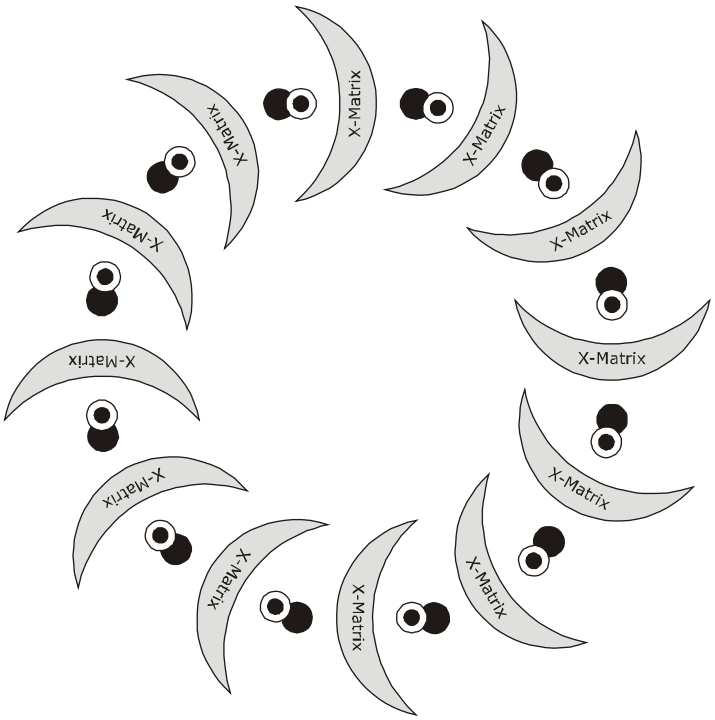
```
X :=
[
X-Matrix:1*ALPHA := {X-Matrix:2*OMEGA, X-Matrix:3*OMEGA};
X-Matrix:2*ALPHA := {X-Matrix:1*OMEGA, X-Matrix:3*OMEGA};
X-Matrix:3*ALPHA := {X-Matrix:1*OMEGA, X-Matrix:2*OMEGA};
X*(p+01) := CX-Matrix:1*p;
X*(p+06) := CX-Matrix:2*p;
X*(p+11) := CX-Matrix:3*p;
|$.X| := 3*(|$.X-Matrix| + 2);
];
```

Wir setzen:
● := X*OMEGA; ⦿ := X*ALPHA;
und können so auf gerichtete Kanten oder Pfeilspitzen im Graphen verzichten, da ja immer gilt:
X*OMEGA :< X*ALPHA;
Jetzt werden schön gestaltete Schaltungen mit Monaden möglich. Und in der Tat verhalten diese sich ähnlich wie einzelne Genien in primären Matrixmengen. X erzeugt nun sogenannte monaoide Sternwandermengen.



Symbolischer Graph von $.X

```
starBUG=3.X:
15.[OP=index.X: X*p := X*p:norm++] =
[
OP=01.X: X*01 :=
3.X-Matrix:1*ALPHA = {3.X-Matrix:2*OMEGA, 3.X-Matrix:3*OMEGA} =
{X*10, X*15};

OP=02.X: X*02 :=
3.CX-Matrix:1*1 = {3.X-Matrix:1*ALPHA, 3.CX-Matrix:1*2, 3.CX-Matrix:1*3} =
{X*01, X*03, X*04};

OP=03.X: X*03 :=
3.CX-Matrix:1*2 = {3.X-Matrix:1*ALPHA, 3.CX-Matrix:1*1, 3.CX-Matrix:1*3} =
{X*01, X*02, X*04};

OP=04.X: X*04 :=
3.CX-Matrix:1*3 = {3.X-Matrix:1*ALPHA, 3.CX-Matrix:1*1, 3.CX-Matrix:1*2} =
{X*01, X*02, X*03};

OP=05.X: X*05 :=
3.X-Matrix:1*OMEGA = {3.CX-Matrix:1*1, 3.CX-Matrix:1*2, 3.CX-Matrix:1*3} =
{X*02, X*03, X*04};

OP=06.X: X*06 :=
3.X-Matrix:2*ALPHA = {3.X-Matrix:1*OMEGA, 3.X-Matrix:3*OMEGA} =
{X*05, X*15};

OP=07.X: X*07 :=
3.CX-Matrix:2*1 = {3.X-Matrix:2*ALPHA, 3.CX-Matrix:2*2, 3.CX-Matrix:2*3} =
{X*06, X*08, X*09};

OP=08.X: X*08 :=
3.CX-Matrix:2*2 = {3.X-Matrix:2*ALPHA, 3.CX-Matrix:2*1, 3.CX-Matrix:2*3} =
{X*06, X*07, X*09};

OP=09.X: X*09 :=
3.CX-Matrix:2*3 = {3.X-Matrix:2*ALPHA, 3.CX-Matrix:2*1, 3.CX-Matrix:2*2} =
{X*06, X*07, X*08};

OP=10.X: X*10 :=
3.X-Matrix:2*OMEGA = {3.CX-Matrix:2*1, 3.CX-Matrix:2*2, 3.CX-Matrix:2*3} =
{X*07, X*08, X*09};

OP=11.X: X*11 :=
3.X-Matrix:3*ALPHA = {3.X-Matrix:1*OMEGA, 3.X-Matrix:2*OMEGA} =
{X*05, X*10};

OP=12.X: X*12 :=
3.CX-Matrix:3*1 = {3.X-Matrix:3*ALPHA, 3.CX-Matrix:3*2, 3.CX-Matrix:3*3} =
{X*11, X*13, X*14};

OP=13.X: X*13 :=
3.CX-Matrix:3*2 = {3.X-Matrix:3*ALPHA, 3.CX-Matrix:3*1, 3.CX-Matrix:3*3} =
{X*11, X*12, X*14};

OP=14.X: X*14 :=
3.CX-Matrix:3*3 = {3.X-Matrix:3*ALPHA, 3.CX-Matrix:3*1, 3.CX-Matrix:3*2} =
{X*11, X*12, X*13};

OP=15.X: X*15 :=
3.X-Matrix:3*OMEGA = {3.CX-Matrix:3*1, 3.CX-Matrix:3*2, 3.CX-Matrix:3*3} =
{X*12, X*13, X*14};
];
```



Ein closed string bestehend aus Monaden der Sorte X-Matrix. Hier darf man sogar auf Kanten im Graph verzichten, wenn man die einzelnen Monadensterne eindeutig zueinander in Beziehung setzt: ◐ ist mit X*OMEGA :< X*ALPHA zu identifizieren.

```
X :=
[
X-Matrix:01*ALPHA := {X-Matrix:12*OMEGA};
X-Matrix:02*ALPHA := {X-Matrix:01*OMEGA};
X-Matrix:03*ALPHA := {X-Matrix:02*OMEGA};
X-Matrix:04*ALPHA := {X-Matrix:03*OMEGA};
X-Matrix:05*ALPHA := {X-Matrix:04*OMEGA};
X-Matrix:06*ALPHA := {X-Matrix:05*OMEGA};
X-Matrix:07*ALPHA := {X-Matrix:06*OMEGA};
X-Matrix:08*ALPHA := {X-Matrix:07*OMEGA};
X-Matrix:09*ALPHA := {X-Matrix:08*OMEGA};
X-Matrix:10*ALPHA := {X-Matrix:09*OMEGA};
X-Matrix:11*ALPHA := {X-Matrix:10*OMEGA};
X-Matrix:12*ALPHA := {X-Matrix:11*OMEGA};
X*(p+001) := CX-Matrix:01*p;
X*(p+015) := CX-Matrix:02*p;
X*(p+029) := CX-Matrix:03*p;
X*(p+043) := CX-Matrix:04*p;
X*(p+057) := CX-Matrix:05*p;
X*(p+071) := CX-Matrix:06*p;
X*(p+085) := CX-Matrix:07*p;
X*(p+099) := CX-Matrix:08*p;
X*(p+113) := CX-Matrix:09*p;
X*(p+127) := CX-Matrix:10*p;
X*(p+141) := CX-Matrix:11*p;
X*(p+155) := CX-Matrix:12*p;
|$.X| := 12*(|$.X-Matrix| + 2);
];
```

X=X-PopMatrix: [$.X := POWSET*$.X-Matrix; |$.X| := |$.X-Matrix|.2*1;];

Beispiele für $=2, $=3 und $=4:

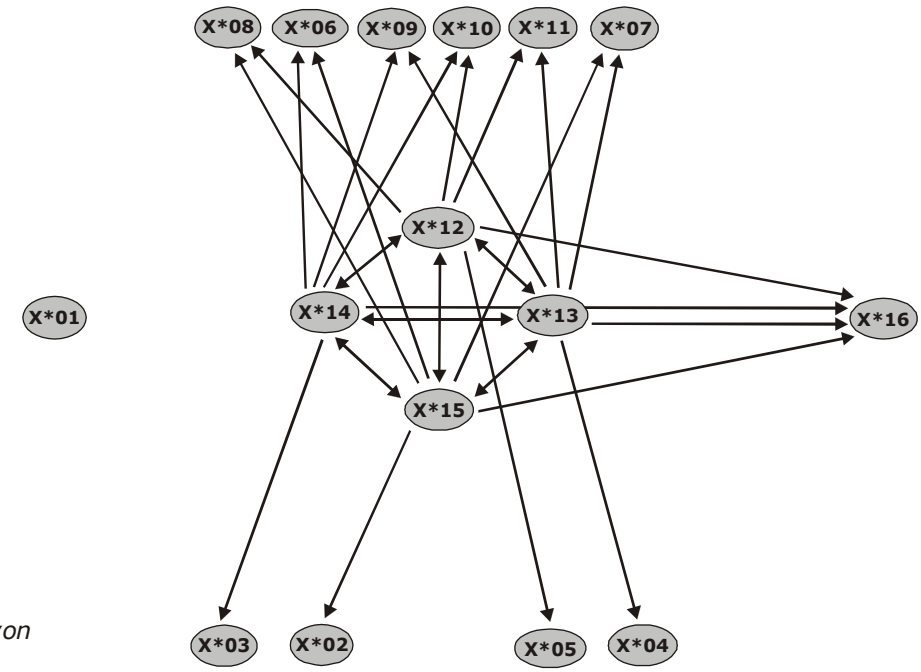
```
starBUG=2.X-Matrix:
2.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2};
OP=2.X: X*2 := {X*1};
];
$2.X-Martix = {X*1, X*2};

$2.X-PopMatrix := POWSET*$2.X-Matrix =
{{ }, {$2.X-Matrix*1}, {$2.X-Matrix*2}, {$2.X-Matrix*1, $2.X-Matrix*2}};

starBUG=2.X-PopMatrix:
4.[OP=index.X: X*p := X*p:norm+] =
[
OP=1.X: X*1 := { } = { };
OP=2.X: X*2 := {$2.X-Matrix*1} = {X*3};
OP=3.X: X*3 := {$2.X-Matrix*2} = {X*2};
OP=4.X: X*4 := {$2.X-Matrix*1, $2.X-Matrix*2} = {X*2, X*3};
];
$2.X-PopMatrix = {X*1, X*2, X*3; X*4};


starBUG=3.X-Matrix=
3.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3};
OP=2.X: X*2 := {X*1, X*3};
OP=3.X: X*3 := {X*1, X*2};
];
$3.X-Matrix = {X*1, X*2, X*3};

$3.X-PopMatrix := POWSET*$3.X-Matrix =
{
{ },
{$3.X-Matrix*1}, {$3.X-Matrix*2}, {$3.X-Matrix*3},
{$3.X-Matrix*1, $3.X-Matrix*2},{$3.X-Matrix*1, $3.X-Matrix*3},
{$3.X-Matrix*2, $3.X-Matrix*3}, {$3.X-Matrix*1, $3.X-Matrix*2, $3.X-Matrix*3}
};

starBUG=3.X-PopMatrix:
8.[OP=index.X: X*p := X*p:norm+] =
[
OP=1.X: X*1 := { } = { };
OP=2.X: X*2 := {$3.X-Matrix*1} = {X*7};
OP=3.X: X*3 := {$3.X-Matrix*2} = {X*6};
OP=4.X: X*4 := {$3.X-Matrix*3} = {X*5};
OP=5.X: X*5 := {$3.X-Matrix*1, $3.X-Matrix*2} = {X*6, X*7};
OP=6.X: X*6 := {$3.X-Matrix*1, $3.X-Matrix*3} = {X*5, X*7};
OP=7.X: X*7 := {$3.X-Matrix*2, $3.X-Matrix*3} = {X*5, X*6};
OP=8.X: X*8 := {$3.X-Matrix*1, $3.X-Matrix*2, $3.X-Matrix*3} = {X*5, X*6, X*7};
];
$3.X-PopMatrix = {X*1, X*2, X*3, X*4, X*5, X*6, X*7, X*8};
```

Hinweis: Mit dem Ausgabetyp norm+ werden im starBUG die Rechenzwischenschritte angezeigt. Will man es aber noch genauer, schreibt man einfach norm++ oder sogar norm+++ usw.

```
starBUG=4.X-Matrix:
4.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4};
OP=2.X: X*2 := {X*1, X*3, X*4};
OP=3.X: X*3 := {X*1, X*2, X*4};
OP=4.X: X*4 := {X*1, X*2, X*3};
];
$4.X-Matrix = {X*1, X*2, X*3, X*4};

$4.X-PopMatrix := POWSET*$4.X-Matrix;
```

Aus Platzgründen und der Übersicht halber lassen wir mal die ausgeschriebene Potenzmenge mit den auf X-Matrix bezogenen Gliedern weg. Man sieht diese Glieder ja auch im unteren starBUG.

```
starBUG=4.X-PopMatrix:
16.[OP=index.X: X*p := X*p:norm+] =
[
OP=01.X: X*01 := { } = { };
OP=02.X: X*02 := {$4.X-Matrix*1} = {X*15};
OP=03.X: X*03 := {$4.X-Matrix*2} = {X*14};
OP=04.X: X*04 := {$4.X-Matrix*3} = {X*13};
OP=05.X: X*05 := {$4.X-Matrix*4} = {X*12};
OP=06.X: X*06 := {$4.X-Matrix*1, $4.X-Matrix*2} = {X*14, X*15};
OP=07.X: X*07 := {$4.X-Matrix*1, $4.X-Matrix*3} = {X*13, X*15};
OP=08.X: X*08 := {$4.X-Matrix*1, $4.X-Matrix*4} = {X*12, X*15};
OP=09.X: X*09 := {$4.X-Matrix*2, $4.X-Matrix*3} = {X*13, X*14};
OP=10.X: X*10 := {$4.X-Matrix*2, $4.X-Matrix*4} = {X*12, X*14};
OP=11.X: X*11 := {$4.X-Matrix*3, $4.X-Matrix*4} = {X*12, X*13};
OP=12.X: X*12 := {$4.X-Matrix*1, $4.X-Matrix*2, $4.X-Matrix*3} = {X*13, X*14, X*15};
OP=13.X: X*13 := {$4.X-Matrix*1, $4.X-Matrix*2, $4.X-Matrix*4} = {X*12, X*14, X*15};
OP=14.X: X*14 := {$4.X-Matrix*1, $4.X-Matrix*3, $4.X-Matrix*4} = {X*12, X*13, X*15};
OP=15.X: X*15 := {$4.X-Matrix*2, $4.X-Matrix*3, $4.X-Matrix*4} = {X*12, X*13, X*14};
OP=16.X: X*16 := {$4.X-Matrix*1, $4.X-Matrix*2, $4.X-Matrix*3, $4.X-Matrix*4} =
{X*12, X*13, X*14, X*15};
];
$4.X-PopMatrix = colX*p;
```



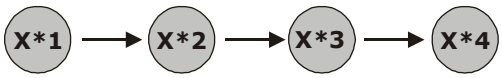
Pfeilediagramm von $4.X-PopMatrix

Es sei nun wieder ein benannter Sternwander X gegeben durch:

X=X-String: [X*(p+1) := X*1 <+ X*p; X*1 := { }; |$.X| := $;];

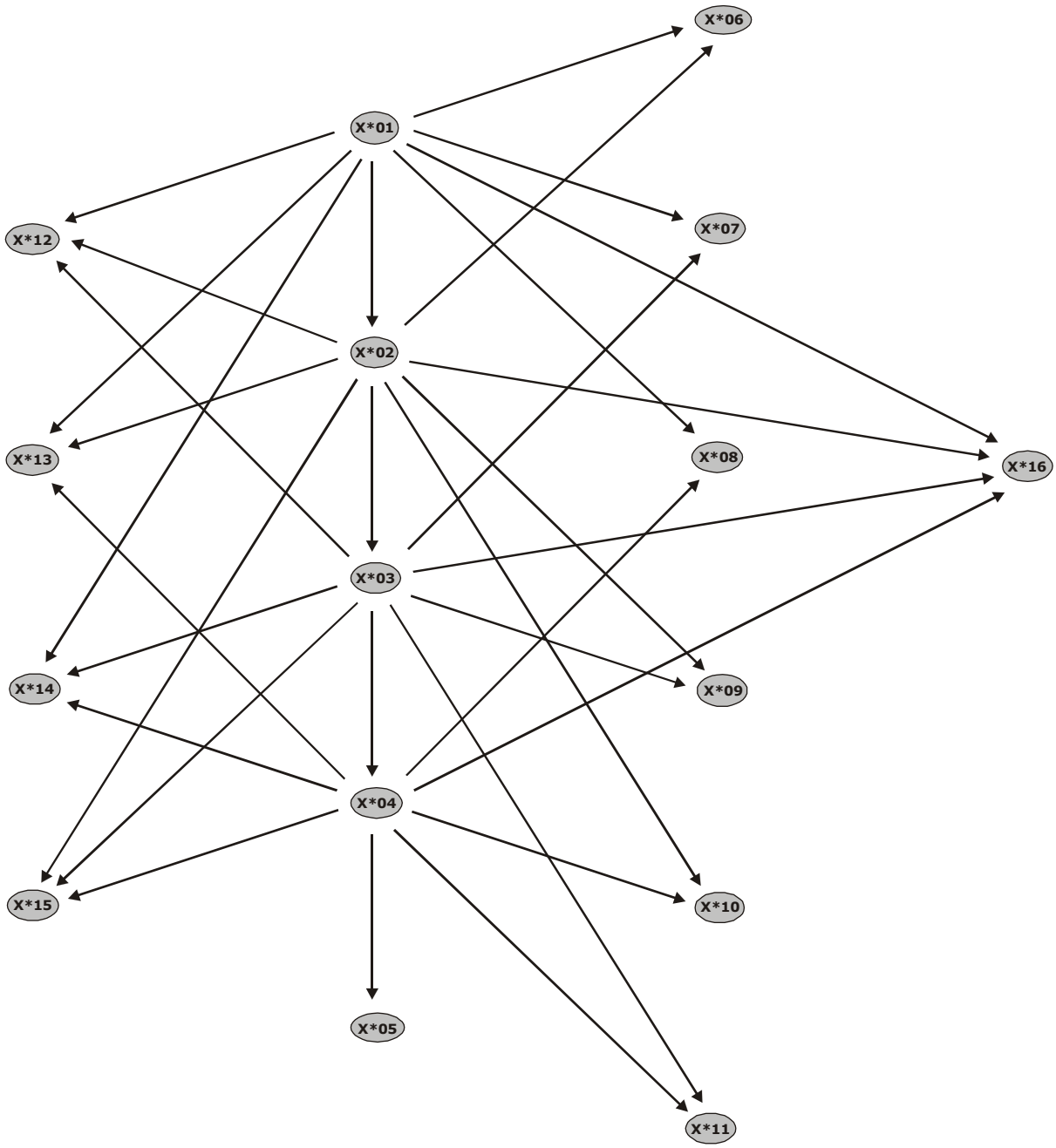
Wir setzen $=4

```
starBUG=4.X-String:
4.[OP=index.X: X*p := X*p:real] =
[
OP=1.X:  X*1 := { };
OP=2.X:  X*2 := {{ }};
OP=3.X:  X*3 := {{{ }}};
OP=4.X:  X*4 := {{{{ }}}};
];

$4.X-String:real = {{ }, {{ }}, {{{ }}}, {{{{ }}}}};
```



Pfeilediagramm von $4.X-String

```
X=X-PopString: [$.X := POWSET*$.X-String; |$.X| := |$.X-String|.2*1;];
=>
$4.X-PopString := POWSET*$4.X-String;

starBUG=4.X-PopString:
16.[OP=index.X: X*p := X*p:norm+] =
[
OP=01.X: X*01 := { } = { };
OP=02.X: X*02 := {{ }} = {X*01};
OP=03.X: X*03 := {{{ }}} = {X*02};
OP=04.X: X*04 := {{{{ }}}} = {X*03};
OP=05.X: X*05 := {{{{{ }}}}} = {X*04};
OP=06.X: X*06 := {{ }, {{ }}} = {X*01, X*02};
OP=07.X: X*07 := {{ }, {{{ }}}} = {X*01, X*03};
OP=08.X: X*08 := {{ }, {{{{ }}}}} = {X*01, X*04};
OP=09.X: X*09 := {{{ }}, {{{ }}}} = {X*02, X*03};
OP=10.X: X*10 := {{{ }}, {{{{ }}}}} = {X*02, X*04};
OP=11.X: X*11 := {{{{ }}}, {{{{ }}}}} = {X*03, X*04};
OP=12.X: X*12 := {{ }, {{ }}, {{{ }}}} = {X*01, X*02, X*03};
OP=13.X: X*13 := {{ }, {{ }}, {{{{ }}}}} = {X*01, X*02, X*04};
OP=14.X: X*14 := {{ }, {{{ }}}, {{{{ }}}}} = {X*01, X*03, X*04};
OP=15.X: X*15 := {{{ }}, {{{ }}}, {{{{ }}}}} = {X*02, X*03, X*04};
OP=16.X: X*16 := {{ }, {{ }}, {{{ }}}, {{{{}}}}} = {X*01, X*02, X*03, X*04};
];
$4.X-PopString = colX*p;
```



Pfeilediagramm von $4.X-PopString

Es sei nun wieder ein benannter Sternwander X gegeben durch:

```
X=X-Space: [X*(p+1) := X*p <+ X*p;  X*1 := { }; |$.X| := $;];
```

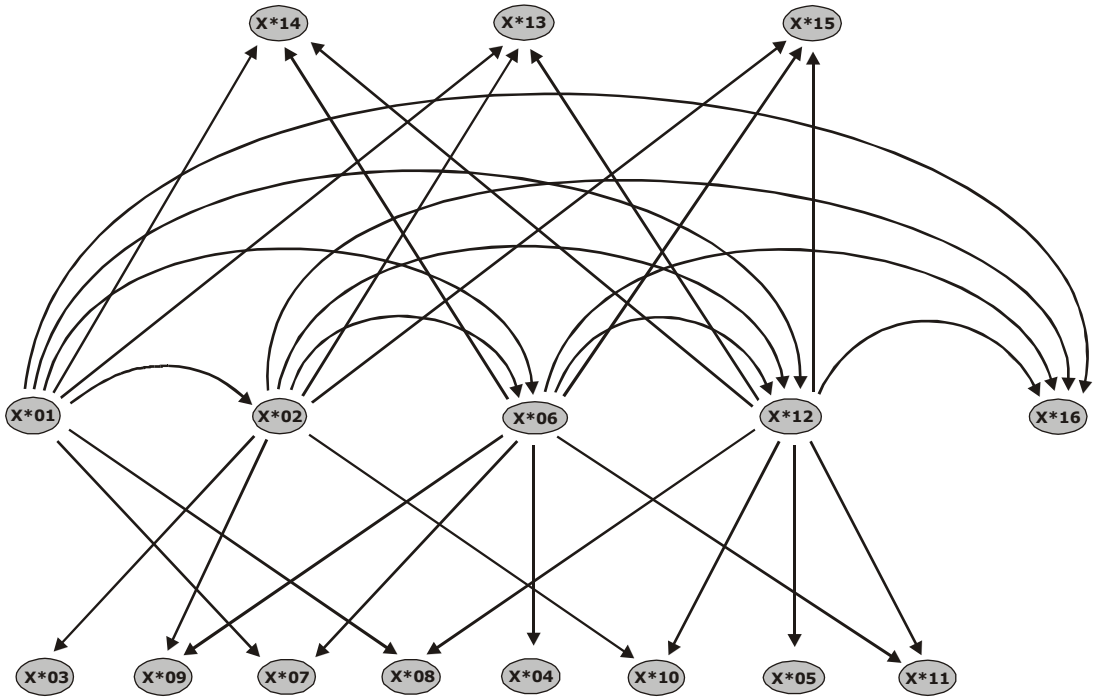
Wir setzen $=4

```
starBUG=4.X-Space:
4.[OP=index.X: X*p := X*p:real] =
[
OP=1.X:  X*1 := { };
OP=2.X:  X*2 := {{ }};
OP=3.X:  X*3 := {{ }, {{ }}};
OP=4.X:  X*4 := {{ }, {{ }}, {{ }, {{ }}}};
];

$4.X-Space = {X*1, X*2, X*3, X*4};
```

```
X=X-PopSpace: [$.X := POWSET*$.X-Space; |$.X| := |$.X-Space|.2*1;];
=>
$4.X-PopSpace := POWSET*$4.X-Space;

starBUG=4.X-PopSpace:
16.[OP=index.X: X*p := X*p:norm++] =
[
OP=01.X: X*01 := { } = { };
OP=02.X: X*02 := {$4.X-Space*1} = {{ }} = {X*01};
OP=03.X: X*03 := {$4.X-Space*2} = {{{ }}} = {X*02};
OP=04.X: X*04 := {$4.X-Space*3} = {{{ }, {{ }}}} = {{X*01, X*02}} = {X*06};
OP=05.X: X*05 := {$4.X-Space*4} = {{{ }, {{ }}, {{ }, {{ }}}}} =
                 {{X*01, X*02, {X*01, X*02}}} =
                 {{X*01, X*02, X*06}} = {X*12};
OP=06.X: X*06 := {$4.X-Space*1, $4.X-Space*2} = {{ }, {{ }}} = {X*01, X*02};
OP=07.X: X*07 := {$4.X-Space*1, $4.X-Space*3} = {{ }, {{ }, {{ }}}} =
                 {X*01, {X*01, X*02}} = {X*01, X*06};
OP=08.X: X*08 := {$4.X-Space*1, $4.X-Space*4} = {{ }, {{ }, {{ }}, {{ }, {{ }}}}} =
                 {X*01, {X*01, X*02, X*06}} = {X*01, X*12};
OP=09.X: X*09 := {$4.X-Space*2, $4.X-Space*3} = {{{ }}, {{ }, {{ }}}} = {X*02, X*06};
OP=10.X: X*10 := {$4.X-Space*2, $4.X-Space*4} = {{{ }}, {{ }, {{ }}, {{ }, {{ }}}}} =
                 {X*02, {X*01, X*02, X*06}} = {X*02, X*12};
OP=11.X: X*11 := {$4.X-Space*3, $4.X-Space*4} =
                 {{{ }, {{ }}}, {{ }, {{ }}, {{ }, {{ }}}}} = {X*06, {X*01, X*02, X*06}} =
                 {X*06, X*12};
OP=12.X: X*12 := {$4.X-Space*1, $4.X-Space*2, $4.X-Space*3} =
                 {{ }, {{ }}, {{ }, {{ }}}} = {X*01, X*02, X*06};
OP=13.X: X*13 := {$4.X-Space*1, $4.X-Space*2, $4.X-Space*4} =
                 {{ }, {{ }}, {{ }, {{ }}, {{ }, {{ }}}}} = {X*01, X*02, X*12};
OP=14.X: X*14 := {$4.X-Space*1, $4.X-Space*3, $4.X-Space*4} =
                 {{ }, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {{ }}}}} = {X*01, X*06, X*12};
OP=15.X: X*15 := {$4.X-Space*2, $4.X-Space*3, $4.X-Space*4} =
                 {{{ }}, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {{ }}}}} = {X*02, X*06, X*12};
OP=16.X: X*16 := {$4.X-Space*1, $4.X-Space*2, $4.X-Space*3, $4.X-Space*4} =
                 {{ }, {{ }}, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {{ }}}}} =
                 {X*01, X*02, X*06, X*12};
];
$4.X-PopSpace = colX*p;
```

```
starBUG=4.X-PopSpace:
16.[OP=index.X: X*p := X*p:norm] =
[
OP=01.X: X*01 := { } = { };
OP=02.X: X*02 := {X*01};
OP=03.X: X*03 := {X*02};
OP=04.X: X*04 := {X*06};
OP=05.X: X*05 := {X*12};
OP=06.X: X*06 := {X*01, X*02};
OP=07.X: X*07 := {X*01, X*06};
OP=08.X: X*08 := {X*01, X*12};
OP=09.X: X*09 := {X*02, X*06};
OP=10.X: X*10 := {X*02, X*12};
OP=11.X: X*11 := {X*06, X*12};
OP=12.X: X*12 := {X*01, X*02, X*06};
OP=13.X: X*13 := {X*01, X*02, X*12};
OP=14.X: X*14 := {X*01, X*06, X*12};
OP=15.X: X*15 := {X*02, X*06, X*12};
OP=16.X: X*16 := {X*01, X*02, X*06, X*12};
];
$4.X-PopSpace = colX*p;
```

Hier wird deutlich, daß doch einige Umformungen notwendig sind, um die Menge $4.X-PopSpace in Orthonormalform darzustellen. Zuerst werden die einzelnen Teilmengen von $4.X-Space in der von X-Space abhängigen Form gebildet und in die Realform umgewandelt, durch die man sukzessiv schließlich alle Terme in die von nur X-PopSpace abhängige Form überführt.
Nach und nach lassen sich alle Mengenausdrücke durch eine entsprechende Substitution mur als reine Teilmengen der Menge $4.X-PopSpace darstellen. Damit ist die noptronische Konsistenz und die Vollständigkeit von $4.X-PopSpace bewiesen.



Pfeilediagramm von $4.X-PopSpace

Menno, was für ein fluffiger Mengen-HokusPokus, wenn das der Bertrand noch erlebt hätte! Auch hier sieht man sehr schön wie die Struktur der gepopten Menge als Noptronische Teilmenge erhalten bleibt. In diesem Beispiel die von Neumann-Zahlen (X-Space) sogar als $(n+1).X-Space bei $n.X-PopSpace. Gilt das also für alle Gradationen von X-PopSpace?

DEC := [$(n+1).X-Space <! $n.X-PopSpace];

Und auch bei $.X-PopMatrix und $.X-String erkennt man diese als Noptronische Teilmenge:

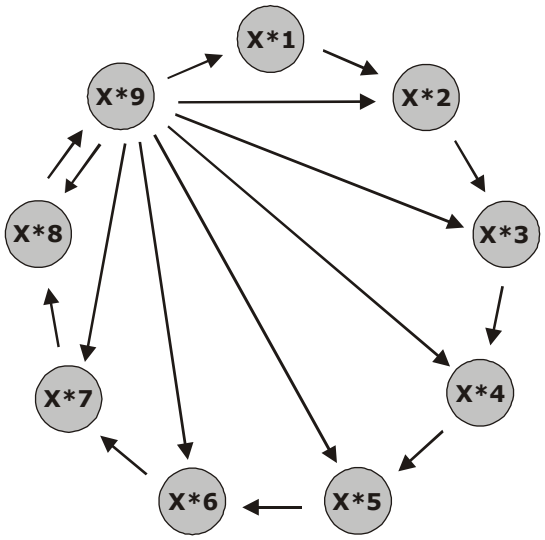
DEC := [$.X-Matrix <! $.X-PopMatrix; $(n+1).X-String <! $n.X-PopString;];

Hier ein schon bekannter Sternwander in einer kreisförmigen Darstellung, was aber natürlich die Menge nicht verändert aber eine andere optische Wirkung hervorruft. Die Verbindungen zwischen den Elementen könnte man sich auch als Gummiseile vorstellen. Jeder Graf den ich so erzeugen kann ohne Verbindungen zu unterbrechen (cut) oder neu zu erzeugen (click) beschreibt die gleiche Matrixmenge. Einigt man sich auf eine kreisförmige Darstellung könnte man in der Tat von "Sternenkreisen" sprechen.

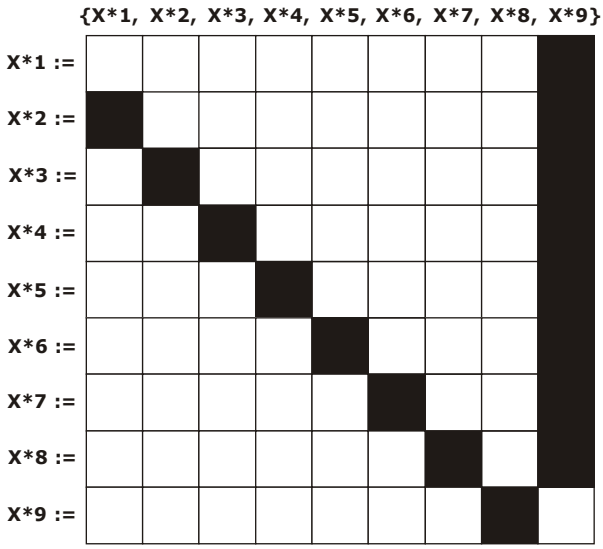
```
X :=
[
X*1 := {X*n};
[(p+1) < n] => [X*(p+1) := X*1 <+ X*p];
[n>1] => [X*n := {X*(n-1)}];
|$.X| := $; n=|$.X|;
];
```

Wir setzen $=9 und lassen uns diesmal die Menge gleich in Orthonormalform anzeigen, denn es handelt sich um eine Imaginäre Sternwandermenge:

```
starBUG=9.X:
9.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*9};
OP=2.X: X*2 := {X*9, X*1};
OP=3.X: X*3 := {X*9, X*2};
OP=4.X: X*4 := {X*9, X*3};
OP=5.X: X*5 := {X*9, X*4};
OP=6.X: X*6 := {X*9, X*5};
OP=7.X: X*7 := {X*9, X*6};
OP=8.X: X*8 := {X*9, X*7};
OP=9.X: X*9 := {X*8};
];
```

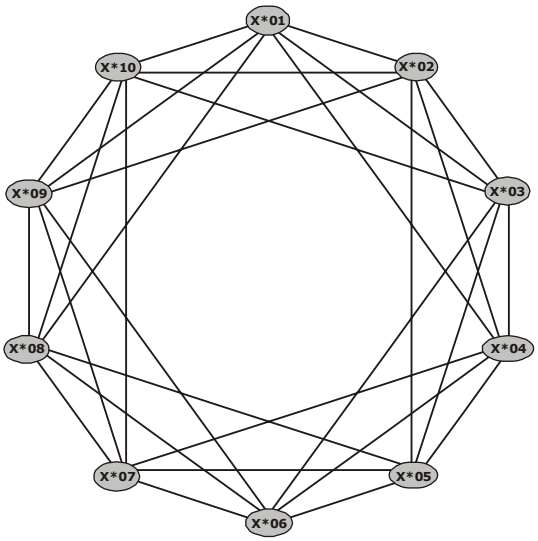


Pfeilediagramm als Sternenkreis

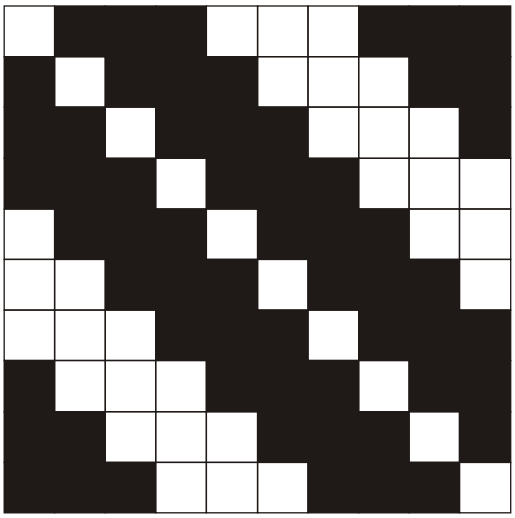


Binärmatrix der Menge $9.X

```
X := [user];

starBUG=10.X:
10.[OP=index.X: X*p := X*p:norm] =
[
OP=01.X: X*01 := {X*02, X*03, X*04, X*08, X*09, X*10};
OP=02.X: X*02 := {X*01, X*03, X*04, X*05, X*09, X*10};
OP=03.X: X*03 := {X*01, X*02, X*04, X*05, X*06, X*10};
OP=04.X: X*04 := {X*01, X*02, X*03, X*05, X*06, X*07};
OP=05.X: X*05 := {X*02, X*03, X*04, X*06, X*07, X*08};
OP=06.X: X*06 := {X*03, X*04, X*05, X*07, X*08, X*09};
OP=07.X: X*07 := {X*04, X*05, X*06, X*08, X*09, X*10};
OP=08.X: X*08 := {X*01, X*05, X*06, X*07, X*09, X*10};
OP=09.X: X*09 := {X*01, X*02, X*06, X*07, X*08, X*10};
OP=10.X: X*10 := {X*01, X*02, X*03, X*07, X*08, X*09};
];
```



Pfeilediagramm als Sternenkreis

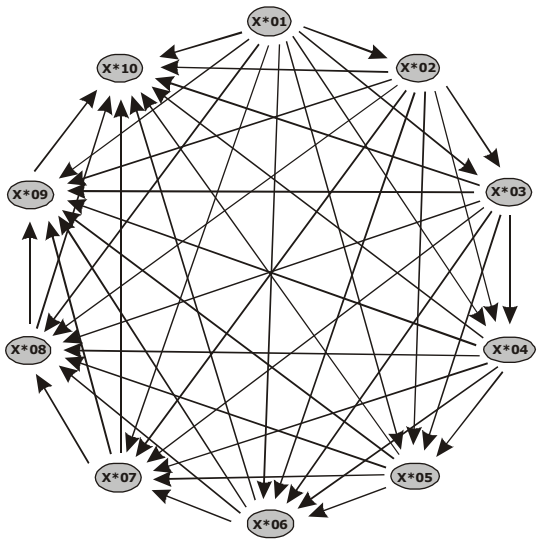


Binärmatrix der Menge $10.X

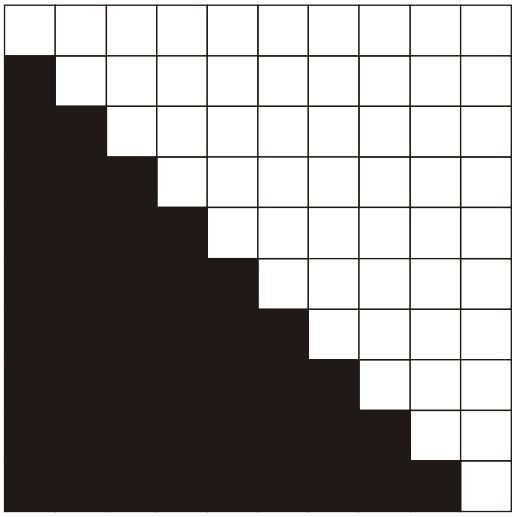
Aufgabe: Finden Sie einen Algorithmus der $.X erzeugt! Was fällt auf wenn man aufeinander folgende Gradationen miteinander vergleicht? Beweis!

(Fauler Sack, mach's doch selber! Anmerkung der Redaktion)

Hier der Sternwander X-Space oder die von Neumann-Zahlen als "Sternenkreis":



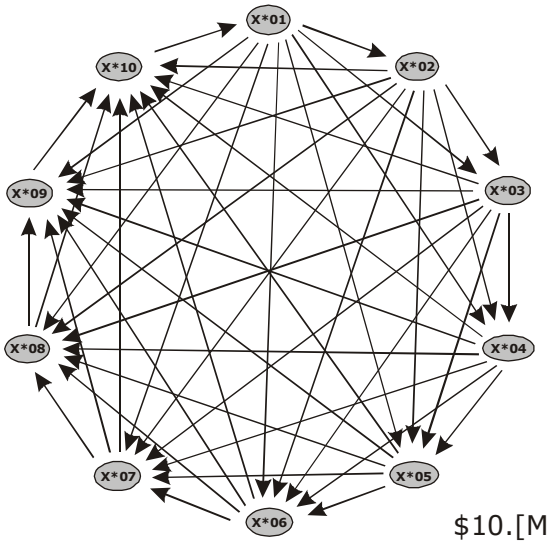
$10.X



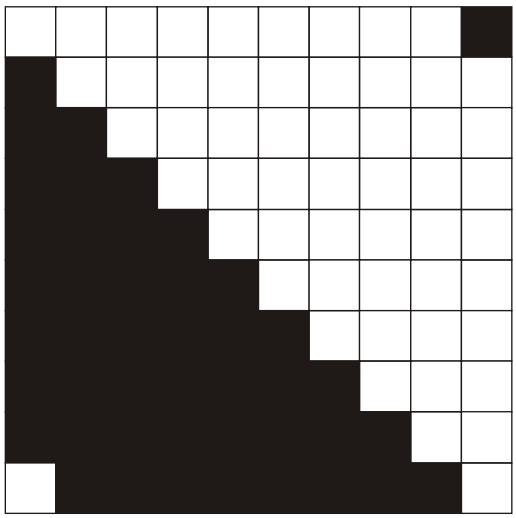
Wir definieren folgenden POP und wenden ihn auf X-Space an:

POP := [OP=1.POP: X*10 <- X*1; X*1 <+ X*10;];

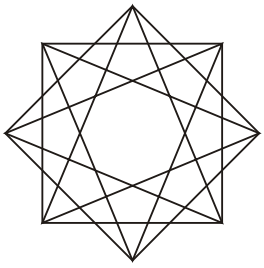
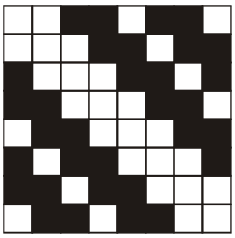
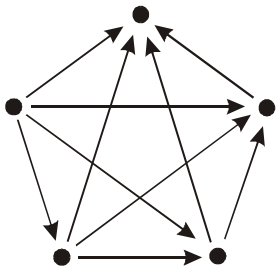
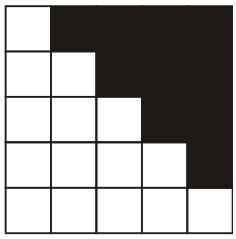
Damit wird X-Space*10 zum Omega-Stern und X-Space*01 zum Alpha-Stern und die gepopte Matrixmenge ist mit einer gültigen Reflexiven Monade identisch:
$10.[POP*X-Space] = $8.C°X-Space;

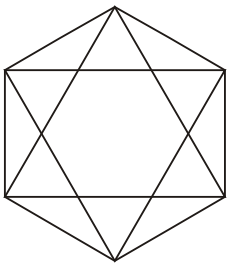
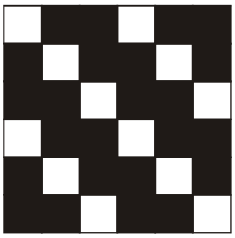
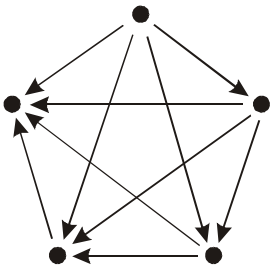
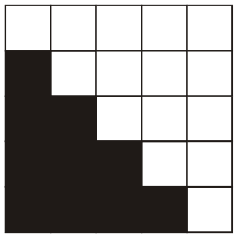


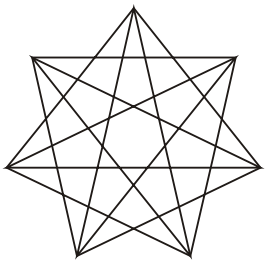
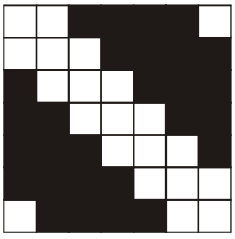
$10.[Moon*X-Space]



Oder etwas allgemeiner formuliert: POP=Moon: [OP=1.POP: X*max <- X*1; X*1 <+ X*max;];
Dann gilt: $.[Moon*X-Space] = ($-2).C°X-Space (wobei C° eine Reflexivmonade bezeichnet)

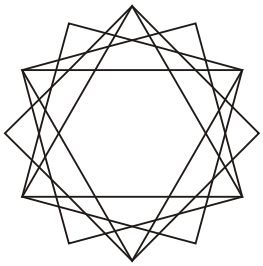
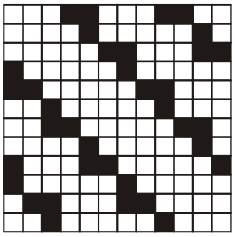
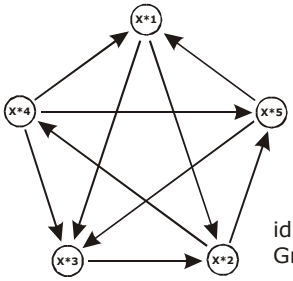





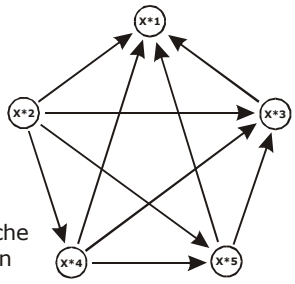



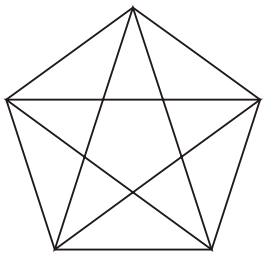
Noptronische Äquivalenzoperation:

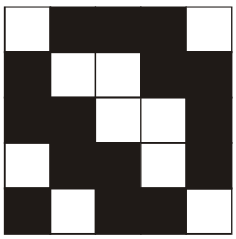
Wird die Binärmatrix horizontal und vertikal gespiegelt, bleibt die Menge gleich. Werden zwei Spalten getauscht und die entsprechenden Zeilen, bleibt die Menge gleich. Untenstehend zwei identische Mengen in unterschiedlicher Graphendarstellung mit einem symmetrischen und einem asymmetrischen c-String als Teilraum.

identische Graphen



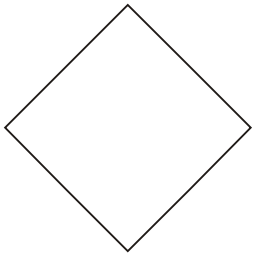
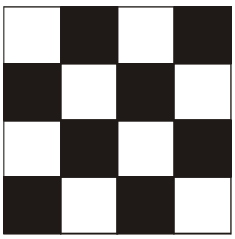
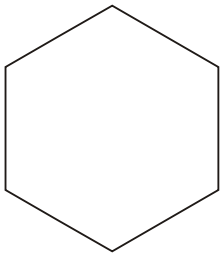
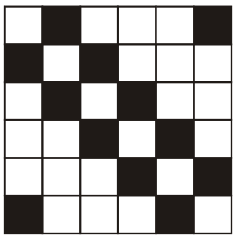




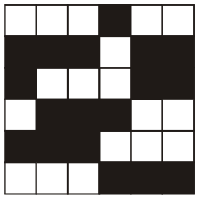
X := [user];

starBUG=5.X:
5.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4};
OP=2.X: X*2 := {X*1, X*4, X*5};
OP=3.X: X*3 := {X*1, X*2, X*5};
OP=4.X: X*4 := {X*2, X*3, X*5};
OP=5.X: X*5 := {X*1, X*3, X*4};
];

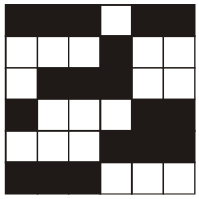
5-Zell, denn X-Matrix modelliert Tetraeder in der Dimension $-1

```
X := [user];

starBUG=6.X:
6.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*4};
OP=2.X: X*2 := {X*1, X*2, X*3, X*5, X*6};
OP=3.X: X*3 := {X*1, X*5, X*6};
OP=4.X: X*4 := {X*2, X*3, X*4};
OP=5.X: X*5 := {X*1, X*2, X*3};
OP=6.X: X*6 := {X*4, X*5, X*6};
];

starBUG=6.X':
6.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*1, X*2, X*3, X*5, X*6};
OP=2.X: X*2 := {X*4};
OP=3.X: X*3 := {X*2, X*3, X*4};
OP=4.X: X*4 := {X*1, X*5, X*6};
OP=5.X: X*5 := {X*4, X*5, X*6};
OP=6.X: X*6 := {X*1, X*2, X*3};
];
```

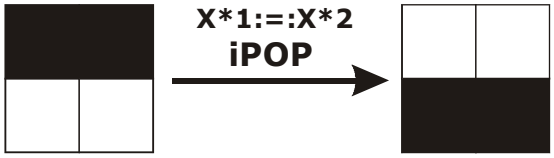


$6.X

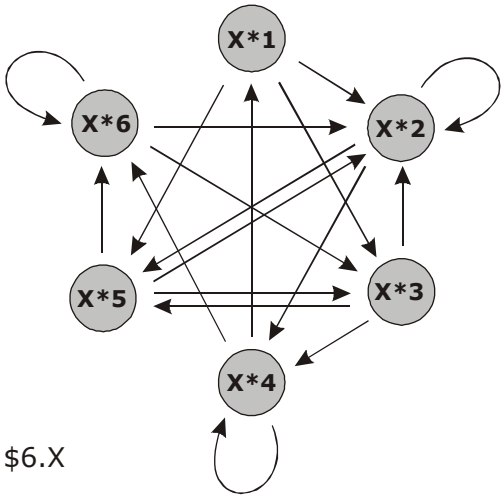


$6.X'

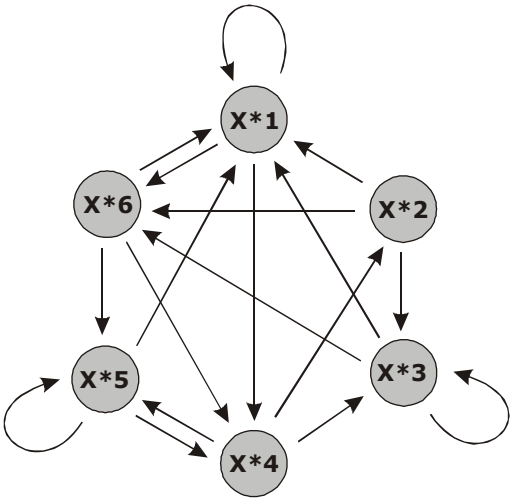
Jede unterliegende Zeile ist komplementär zur oberen gewählt. Die Idee:
Eine Noptronische Menge zu erzeugen, deren Komplementärmenge zu dieser identisch ist. Die Menge $6.X ist zwar nicht identisch mit $6.X' aber hier ist eine ganz einfache Lösung:



$2.X $2.X'



$6.X



$6.X'

Wie kann man zeigen, daß $6.X # $6.X' ist? Da beide äquinoptisch sind, ist dies nicht sofort ersichtlich: NOPOS*$6.X = NOPOS*$6.X' = (18:6);
Gilt für alle Sternwander $.X # $.X' oder existiert ein Sternwander X für den gilt: $.X = $.X' ?

Möglicher Beweis:
$6.X' hat zwei reflexive, einfach symmetrische Elemente (X*1; X*5;) während $6.X nur genau ein reflexives, einfach symmetrisches Element (X*2) hat.
Wie kann man entscheiden ob Noptronische Mengen unterschiedlich sind ohne alle möglichen iPOPs durchzurechnen? Die entsprechende Rechenzeit wird mit der Mächtigkeit der Matrixmenge sehr schnell zunehmen (NP?). Inwieweit ist die Verschiedenheit überhaupt algorithmisch zu entscheiden?
Selbst für $1.X = {{ }} gilt $1.X # $1.X' wie man leicht sieht, denn das Element X*1 aus $1.X ist in seinem starBUG so definiert X*1 := { }; während das Element X*1 aus $1.X' diese Definition hat: X*1 := {X*1}; und sich damit die Matrixmengen $1.X und $1.X' voneinander unterscheiden.

Hinweis: X' wird wie ein benamster Sternwander behandelt! Werden die Glieder aus einem Zusammenhang (starBUG) entfernt muß man genauso einen (') setzen wie auch einen eventuell gegebenen Namen anfügen. Um einen X' zu definieren muß man jedoch zuerst X definieren, kann dann aber ohne weiteres den starBUG von X' ausführen lassen.

Man könnte beispielsweise deltaCHECK einsetzen. Diese Programm untersucht alle möglichen Paare von Elementen auf ihren Abstand. Lassen sich die dabei entstehenden Abstände nicht in einer Identitätsrelation zuordnen, unterscheiden sich die Mengen sicher. Existiert diese Zuordnung wie hier in unserem Beispiel, können die Mengen dennoch verschieden sein. Dann kann man noch deltaCHECK auf Teilmengen anwenden und vergleichen, z. B. auf alle Elemente die sich selbst enthalten. Da ihre Anzahl in unseren Mengen die gleiche ist, könnten sie sich aber vielleicht in ihren Abständen zueinander unterscheiden. Ist das auch nicht zielführend, muß man eben versuchen einen typeCHECK so zu konstruieren, daß der Unterschied der beiden Mengen zu Tage tritt.

```
DEF=typeCHECK:
[
var: n;
typeCHECK := n.[OP=index: type];
];
```

```
DEF=deltaCHECK:
[
var: n, p, q; p#q;
deltaCHECK=$.X:
n.[OP=index: {X*p}@{X*q}=delta];
];
```

```
deltaCHECK=$6.X:
30.[OP=index: {X*p}@{X*q}=delta;] =
[
OP=01: {X*1}@{X*2}=2;
OP=02: {X*1}@{X*3}=2;
OP=03: {X*1}@{X*4}=1;
OP=04: {X*1}@{X*5}=3;
OP=05: {X*1}@{X*6}=3;
OP=06: {X*2}@{X*1}=1;
OP=07: {X*2}@{X*3}=1;
OP=08: {X*2}@{X*4}=2;
OP=09: {X*2}@{X*5}=1;
OP=10: {X*2}@{X*6}=1;
OP=11: {X*3}@{X*1}=1;
OP=12: {X*3}@{X*2}=2;
OP=13: {X*3}@{X*4}=2;
OP=14: {X*3}@{X*5}=1;
OP=15: {X*3}@{X*6}=1;
OP=16: {X*4}@{X*1}=2;
OP=17: {X*4}@{X*2}=1;
OP=18: {X*4}@{X*3}=1;
OP=19: {X*4}@{X*5}=2;
OP=20: {X*4}@{X*6}=2;
OP=21: {X*5}@{X*1}=1;
OP=22: {X*5}@{X*2}=1;
OP=23: {X*5}@{X*3}=1;
OP=24: {X*5}@{X*4}=2;
OP=25: {X*5}@{X*6}=2;
OP=26: {X*6}@{X*1}=2;
OP=27: {X*6}@{X*2}=2;
OP=28: {X*6}@{X*3}=2;
OP=29: {X*6}@{X*4}=1;
OP=30: {X*6}@{X*5}=1;
];
```

```
deltaCHECK=$6.X':
30.[OP=index: {X*p}@{X*q}=delta;] =
[
OP=01: {X*1}@{X*2}=1;
OP=02: {X*1}@{X*3}=1;
OP=03: {X*1}@{X*4}=2;
OP=04: {X*1}@{X*5}=1;
OP=05: {X*1}@{X*6}=1;
OP=06: {X*2}@{X*1}=2;
OP=07: {X*2}@{X*3}=3;
OP=08: {X*2}@{X*4}=1;
OP=09: {X*2}@{X*5}=2;
OP=10: {X*2}@{X*6}=2;
OP=11: {X*3}@{X*1}=2;
OP=12: {X*3}@{X*2}=1;
OP=13: {X*3}@{X*4}=1;
OP=14: {X*3}@{X*5}=2;
OP=15: {X*3}@{X*6}=2;
OP=16: {X*4}@{X*1}=1;
OP=17: {X*4}@{X*2}=2;
OP=18: {X*4}@{X*3}=2;
OP=19: {X*4}@{X*5}=1;
OP=20: {X*4}@{X*6}=1;
OP=21: {X*5}@{X*1}=2;
OP=22: {X*5}@{X*2}=2;
OP=23: {X*5}@{X*3}=2;
OP=24: {X*5}@{X*4}=1;
OP=25: {X*5}@{X*6}=1;
OP=26: {X*6}@{X*1}=1;
OP=27: {X*6}@{X*2}=1;
OP=28: {X*6}@{X*3}=1;
OP=29: {X*6}@{X*4}=2;
OP=30: {X*6}@{X*5}=2;
];
```

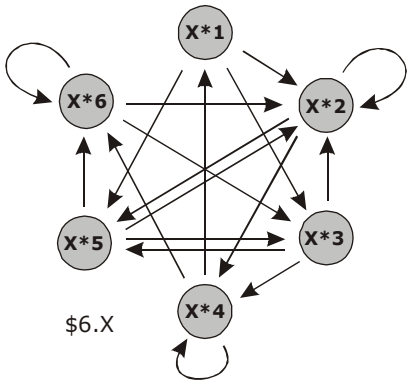
```
$6.partX := colX*p:ref;
deltaCHECK=$6.partX:
6.[OP=index: {X*r}@{X*q}=delta;] =
[
OP=1: {X*2}@{X*4}=2;
OP=2: {X*2}@{X*6}=1;
OP=3: {X*4}@{X*2}=1;
OP=4: {X*4}@{X*6}=2;
OP=5: {X*6}@{X*2}=2;
OP=6: {X*6}@{X*4}=1;
];
```

```
$6.partX' := colX*p:ref;
deltaCHECK=$6.partX':
6.[OP=index: {X*r}@{X*q}=delta;] =
[
OP=1: {X*2}@{X*4}=1;
OP=2: {X*2}@{X*6}=2;
OP=3: {X*4}@{X*2}=2;
OP=4: {X*4}@{X*6}=1;
OP=5: {X*6}@{X*2}=1;
OP=6: {X*6}@{X*4}=2;
];
```

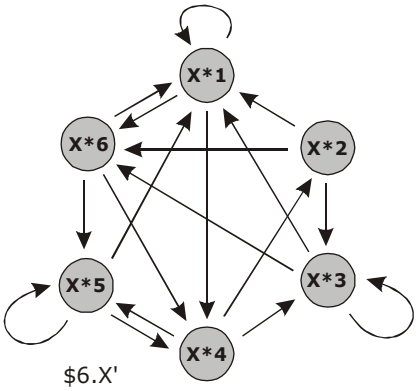
Wie man leicht nachprüfen kann, existiert für beide deltaCHECK eine Identitätsabbildung, denn zu jeder Operation OP=n: {X*p}@{X*q}=d; aus deltaCHECK=$6.X; läßt sich sicher eine Operation OP=m: {X*r}@{X*s}=k; aus deltaCHECK=$6.X'; mit d=k finden. Das Gleiche gilt für die Teilmengen $6.partX und $6.partX'. Bewiesen ist damit aber noch rein gar nichts. In allgemeiner Form sieht das so aus:

[deltaCHECK=$.X<=>deltaCHECK=$.X']: n.[OP=index: {X*p}@{X*q}={X'*r}@{X'*s}];

DEF=ref:
[
type=ref: set :< set;
];



$6.X

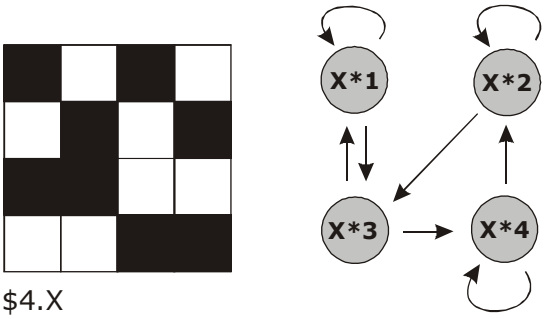


$6.X'

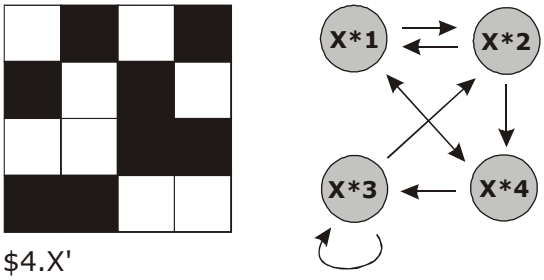
```
type=[{X*p, X*q}:sym & X*(p%q):ref];
typeCHECK=$6.X:
1.[OP=index: {X*p, X*q}:sym & X*(p%q):ref] =
[
OP=1: {X*2, X*5}:sym & X*2:ref];
];
```

```
type=[{X*p, X*q}:sym & X*(p%q):ref];
typeCHECK=$6.X':
2.[OP=index: {X*p, X*q}:sym & X*(p%q):ref] =
[
OP=1: {X*1, X*6}:sym & X*1:ref;
OP=2: {X*4, X*5}:sym & X*5:ref;
];
```

Hier wird die Notwendigkeit einer Beweisidee deutlich. Es stellt sich also die Frage, ob es einen ultimativen Algorithmus gibt, der dies auch leisten kann. Jedoch steht der Verdacht im Raum, daß sich nur die Verschiedenheit von zwei Mengen beweisen läßt ohne alle iPOPs durch zu rechnen. Sind zwei Mengen identisch, kann zum Beweis die entsprechende iPOP-Permutation nur durch Probieren gefunden werden. Und das kann dauern. Besonders bei unendlichen Mengen.
Und hier haben wir eine typeCHECK gefunden der die Verschiedenheit beider Mengen beweist, denn diese Verknüpfung von Eigenschaften hat in jeder Menge eine andere Anzahl von darauf zutreffenden Elementen und daher existiert kein iPOP der die eine Menge in die andere überführt. Damit ist die Gültigkeit von $6.X#$6.X' evident. Fragen wir doch mal den 2MPU, was er meint, denn der hat immer recht, sagt er jedenfalls. DEC := [$6.X#$6.X']; REAL*DEC=1; Sieh'ste Gritli, ich wußte es!



$4.X



$4.X'

$4.X wurde so angelegt, daß immer zwei Zeilen komplementär sind. $4.X' kann also auch erzeugt werden durch Vertauschen der entsprechenden Zeilen der Binärmatrix. Es geht hier um die Frage, ob sich $.X für alle Sternwander immer von $.X' unterscheidet. Hier sieht man sofort $4.X # $4.X' da sicher keine iPOP Umformung aufgrund der unterschiedlichen Anzahl der reflexiven Elemente möglich ist. Hier ein paar kleine noch übersichtliche Beispiele mit einer möglichen Lösung:

$2.X # $2.X'

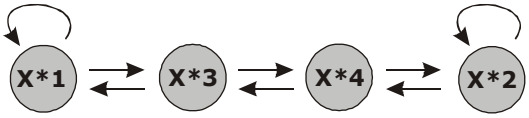
$2.X # $2.X'

$2.X # $2.X'

$2.X = $2.X'

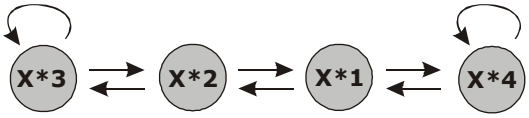
Nun eine Noptronische Menge mit 4 Elementen die tatsächlich mit ihrer eigenen Komplementärmenge identisch ist. Wobei hier nur die Identität einer einzelnen Menge mit ihrem Komplement vorliegt. Interessant ist die Frage ob ein kompletter Sternwander mit dieser Eigenschaft existiert!

```
X := [user];
starBUG=4.X:
4.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*1, X*3};
OP=2.X: X*2 := {X*2, X*4};
OP=3.X: X*3 := {X*1, X*4};
OP=4.X: X*4 := {X*2, X*3};
];
```

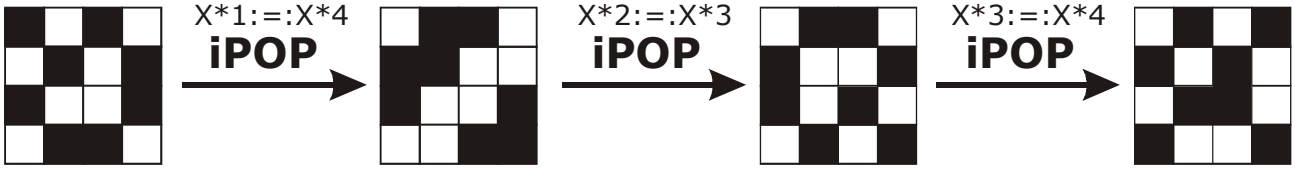


$4.X

```
starBUG=4.X':
4.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*4};
OP=2.X: X*2 := {X*1, X*3};
OP=3.X: X*3 := {X*2, X*3};
OP=4.X: X*4 := {X*1, X*4};
];
```



$4.X'



$4.X

$4.X'

Und in der Tat ist die Identität der Mengen $4.X und $4.X' anhand der Graphendarstellung evident. Die Netzwerkstruktur stimmt überein und $4.X läßt sich mit einem simplen dreistufigen iPOP in $4.X' überführen:

```
iPOP := [OP=1.iPOP: X*1:=:X*4; OP=2.iPOP: X*2:=:X*3; OP=3.iPOP: X*3:=:X*4;];
iPOP*4.X = 4.X'; => $4.X = $4.X';
```

Fig. 1 ist durch die Transitivität der Identitätsrelation sicher richtig.
Vermutung: Dies gilt jedoch auch bei Fig. 2 wenn $.X#$.X';
Es scheint als entstünde bei der Komplementär-Operation ebenso eine Mengenidentität als reine Netzwerkstruktur. Oder anders gesagt, gleiche Netzwerkstrukturen erzeugen zwar formal unterschiedliche Komplementärmengen, aber gleiche Mengen. Lediglich die Netzwerkstruktur bestimmt unserer Noptronischen Menge während die Benennung ihrer Elemente irrelevant ist. Dies bestätigt die Noptronische Mengentheorie als rein kenosemiotischen Ansatz einer weiterführenden Kulturrevolution im Rahmen eines metaorientierten Paradigmenwechsels. Kultur wird zu einem polykontexturellen Phänomen unserer Zeit. Vergangenheit und Zukunft verschmelzen zur Einheit der Gedanken eines einzig ewig einig Volkes. Die Gegenwart gab es nie. Fragen Sie einfach Lexee Smith, die geilste Lolita-Schnecke vom Klondike bis New Mecico.

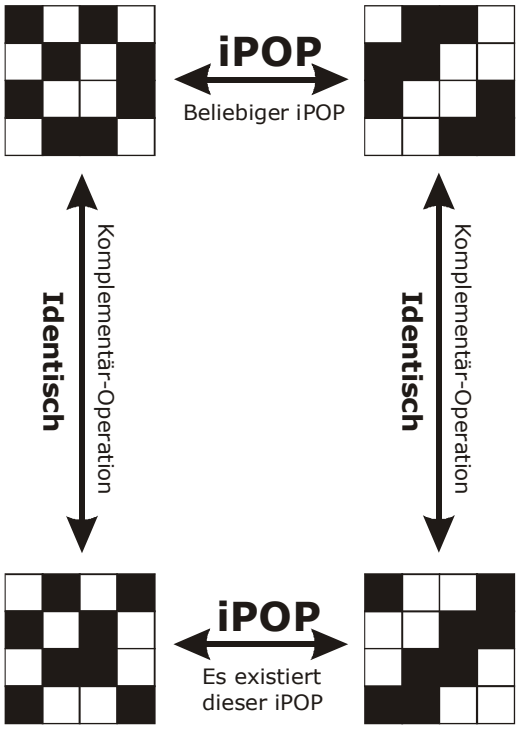


Fig. 1

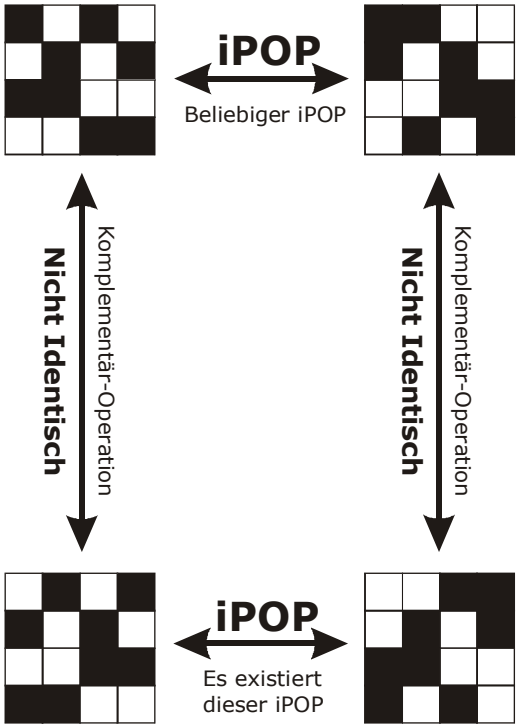
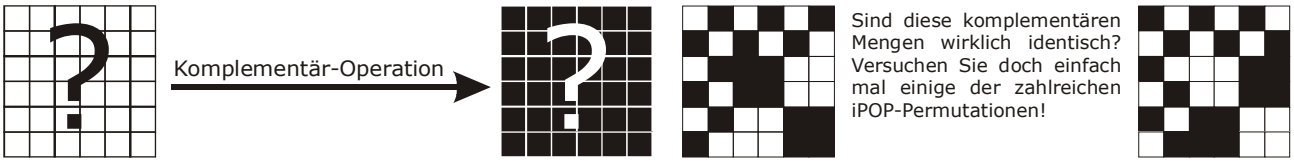


Fig. 2

Fazit: Die Komplementär-Operation einer Noptronischen Menge ist also eine rein mengentheoretische Operation die unabhängig von der formalen Darstellung der Menge operiert. Hier wird einzig die Netzwerkstruktur welche die Menge ja selbst sein soll übertragen. Leider fällt mir grad' keine Idee für einen Beweis beliebiger Mengen ein. Fig. 1 gilt offensichtlich immer, Fig. 2 bedarf eines korrekten Beweises.
Weiter stellen wir uns die Frage ob beliebig viele solcher Fixpunktmengen existieren. Oder sogar ein kompletter Fixpunkt-Sternwander ($.X = $.X') der für alle Gradationen Fixpunktmengen erzeugt und rein algorithmisch funktioniert. Manuelle Bastelei gilt da nicht, denn das macht ja der Benutzer und der hat ein Bewußtsein! Jedoch, schon allein eine Noptronische Menge mit mickrigen 6 Elementen (Mengen mit ungerader Mächtigkeit scheiden aus. Beweis!) als Fixpunktmenge zu konstruieren stellt sich schwierig dar. Die mögliche Anzahl gültiger Binärmatrizen beträgt hier schon (2 hoch 6)!/(2 hoch 6 minus 6)! = 53981544960 Natürlich kommen lediglich Mengen mit der Mannigfaltigkeit $n^2/2$ in Frage, da nur dann überhaupt die Möglichkeit der Identität der beiden Mengen $.X und $.X' besteht. Selbst für n=6 ist es eher knifflig ohne Computerunterstützung eine Fixpunktmenge zu finden. Versuchen Sie es mal!

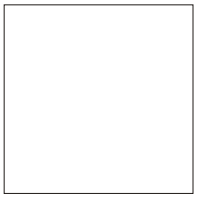


Wir definieren uns folgenden Sternwander der in Fachkreisen als "Fischli-Menge" bekannt ist. Hier wird eine Sternwander durch Manipulation einer Binärmatrix erzeugt. Diese Sternwandermenge wird also als Bit-Kette sukzessiv berechnet und genügt der Guthen Sternwanderey. (Beweis!)
Zuerst aber müssen wir uns die Binärmatrix als noptonisches Objekt über die reine Bitkette stringBIT definieren:

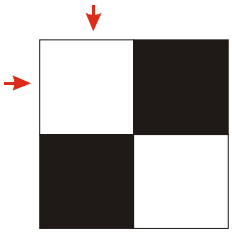
```
DEF=matrix:
matrix=stringBIT:
                    [
                    y:matrix.[zeile := zeile:index];
                    x:matrix.[matrix := <zeile*spalte:index>];
                    ];
```

```
X=X-ping:
[
OP=1.ping:
[
var: z, s;
matrix=(), mark=();
z=zeile;
s=spalte;
|$| := mark:max;
]
OP=2.ping:
[
OP=1: matrix := matrix+z:mark;
OP=2: z := invert*z:max;
OP=3: matrix := matrix+s:mark;
OP=4: s := invert*s:max;
]
OP=4.ping:
[
(test:MQL)*matrix=ERROR; =>
                    [
                    X*new :< X*new; => X*new <- X*new;
                    X*new #:< X*new; => X*new <+ X*new;
                    ];
];
OP=5.ping:
[
OP=1: mark#0; => mark := mark - 1;
OP=2: mark=0; => mark := mark:max;
OP=3: mark:max=$; => [
                         OP=1: WRTLN=matrix;
                         OP=2: OP=0.ping;
                         ];
];
OP=6.ping: OP=2.ping;
];
```

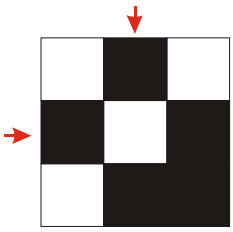
Startwert: matrix=<0>; marker=1;



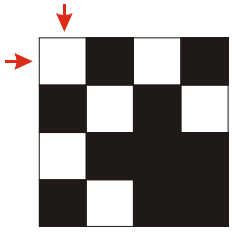
$1.X-ping =
{X*1} = {{ }};



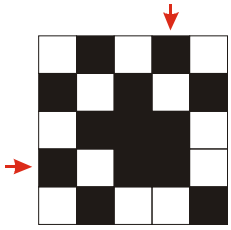
$2.X-ping



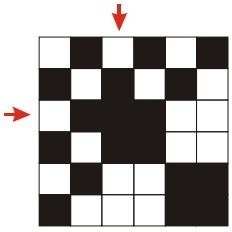
$3.X-ping



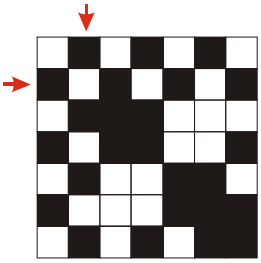
$4.X-ping



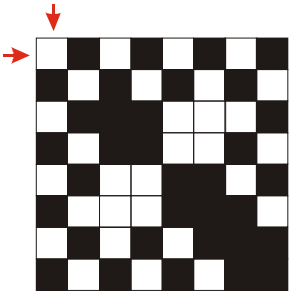
$5.X-ping



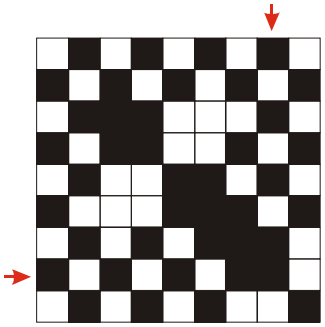
$6.X-ping



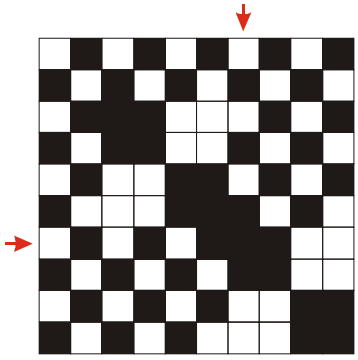
$7.X-ping



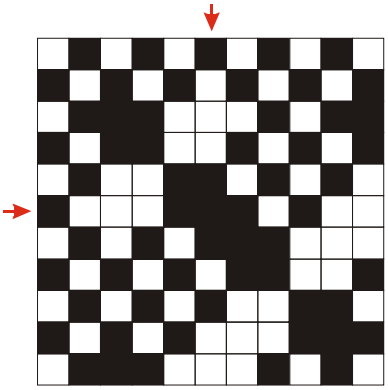
$8.X-ping



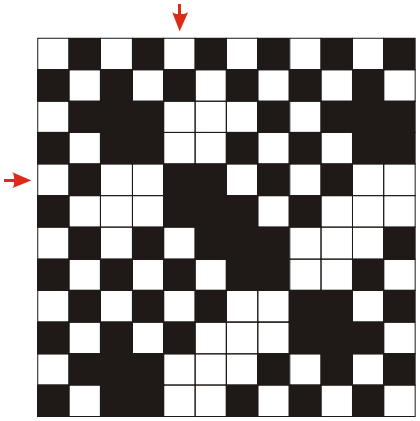
$9.X-ping



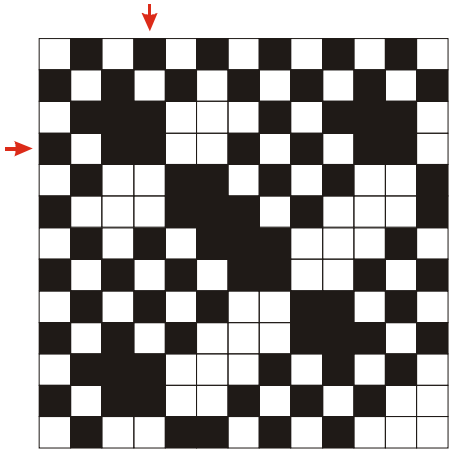
$10.X-ping



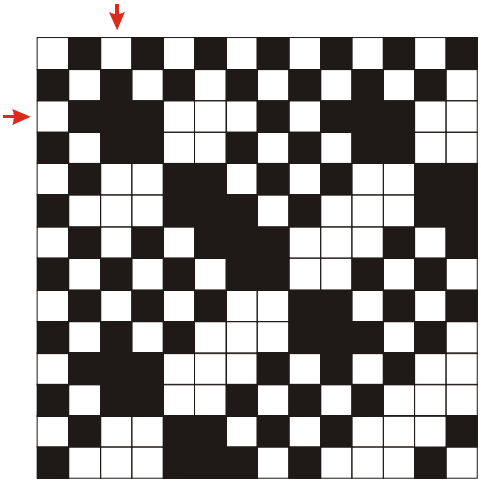
$11.X-ping



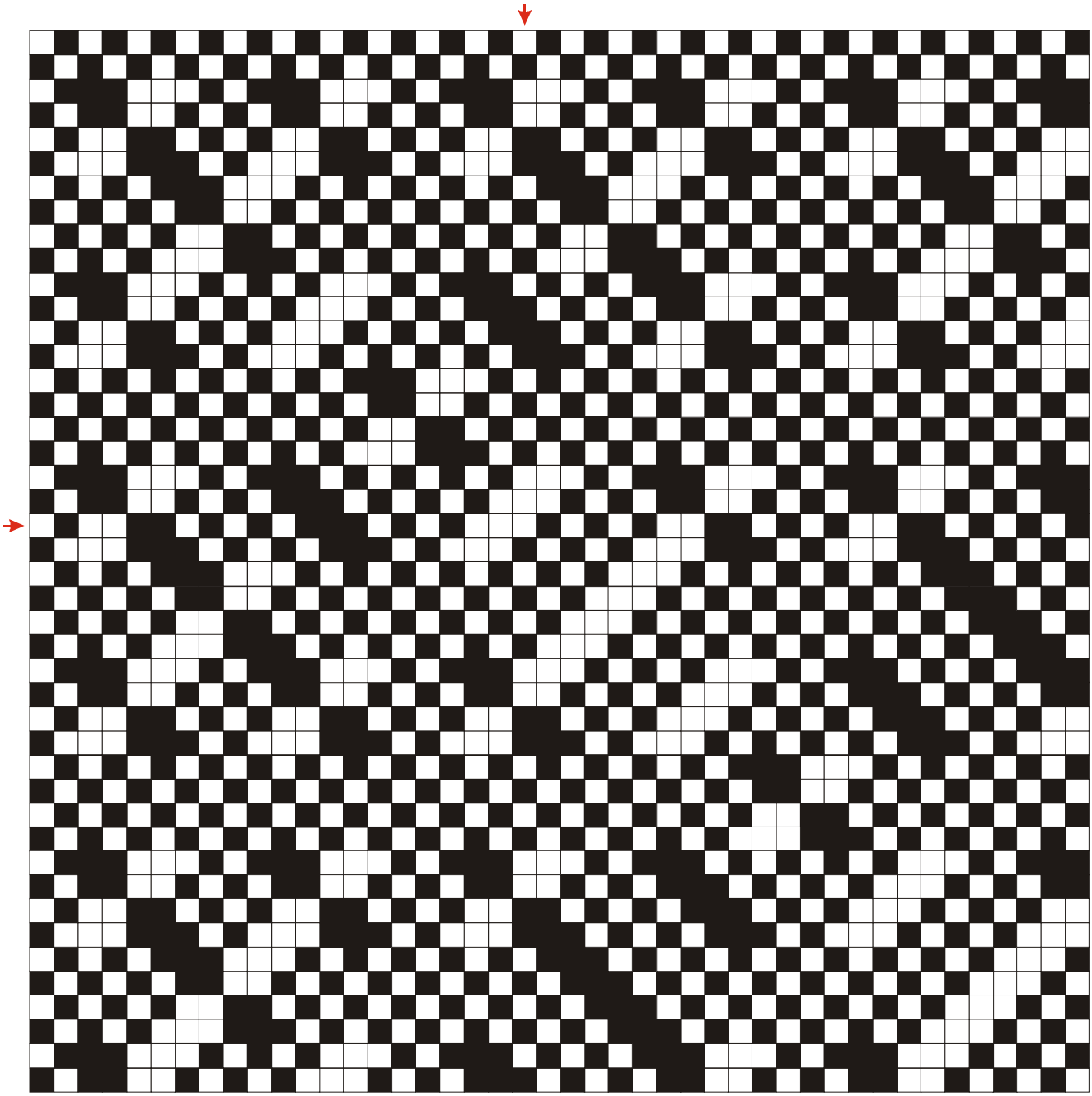
$12.X-ping



$13.X-ping



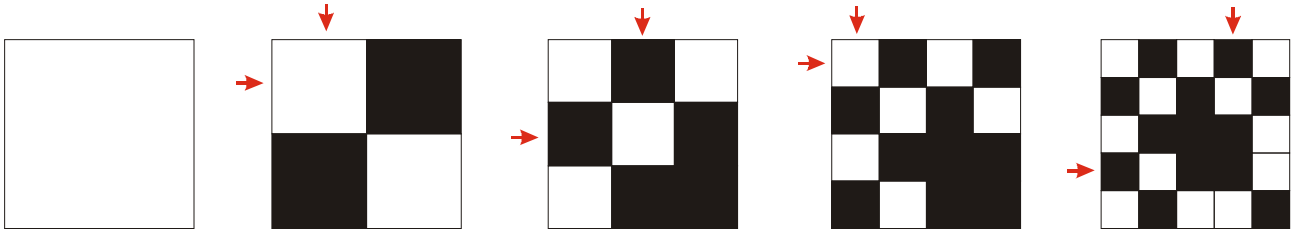
$14.X-ping



$43.X-ping

Ob das Ding wohl irgendwann periodisch wird, sich also mit einer Elementarzelle kacheln läßt? Oder ist es etwa ein Quasikristall? Niemand weiß es und vielleicht werden wir es nie erfahren.

matrix=<0>; marker=1; Man könnte natürlich jede beliebige noptronische Menge als Startwert wählen. Also auch unterschiedliche Binärmatrizen die gleiche Mengen darstellen. Hier stellt sich nun die Frage ob die daraus folgenden Binärmatrizen auch die gleichen Sternwandermengen sind.



$1.X-ping = {X*1} = {{ }};

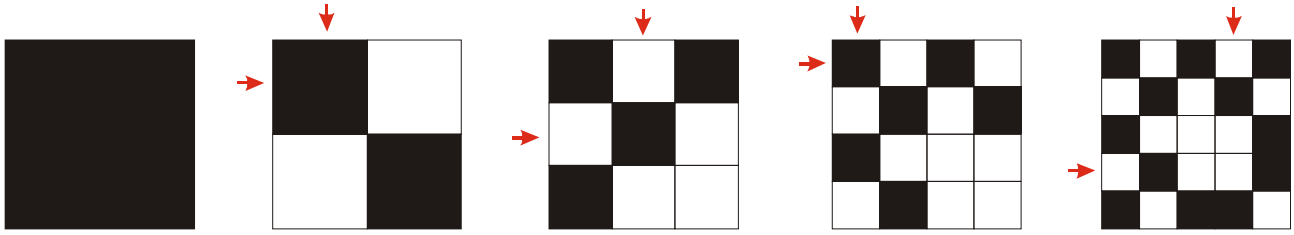
$2.X-ping

$3.X-ping

$4.X-ping

$5.X-ping

Wählt man matrix=<1>; marker=1; ergibt sich der zu X komplementäre Sternwander X':



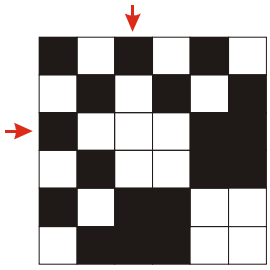
$1.X'-ping = {X*1} = {{X*1}}= omset;

$2.X'-ping

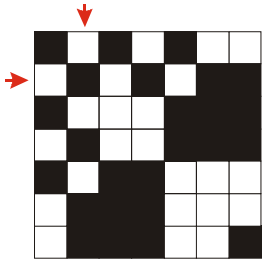
$3.X'-ping

$4.X'-ping

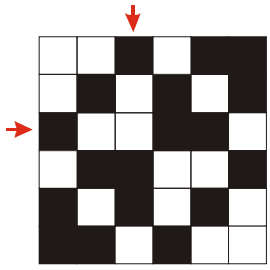
$5.X'-ping



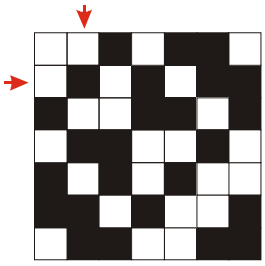
```
X=X-S1: [user];
starBUG=6.X:
6.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*1, X*3, X*5};
OP=2.X: X*2 := {X*2, X*4, X*6};
OP=3.X: X*3 := {X*1, X*5, X*6};
OP=4.X: X*4 := {X*2, X*5, X*6};
OP=5.X: X*5 := {X*1, X*3, X*4};
OP=6.X: X*6 := {X*2, X*3, X*4};
];
```



```
X=X-S3: [ping(mark=3)*($-1).X-S1];
starBUG=7.X:
7.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*1, X*3, X*5};
OP=2.X: X*2 := {X*2, X*4, X*6, X*7};
OP=3.X: X*3 := {X*1, X*5, X*6, X*7};
OP=4.X: X*4 := {X*2, X*5, X*6, X*7};
OP=5.X: X*5 := {X*1, X*3, X*4};
OP=6.X: X*6 := {X*2, X*3, X*4};
OP=7.X: X*7 := {X*2, X*3, X*4, X*7};
];
```



```
X=X-S2:
[
[iPOP := [OP=1.iPOP: X*1 :=: X*3]];
X-S2 := iPOP*X-S1;
];
starBUG=6.X:
6.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*3, X*5, X*6};
OP=2.X: X*2 := {X*2, X*4, X*6};
OP=3.X: X*3 := {X*1, X*4, X*5};
OP=4.X: X*4 := {X*2, X*3, X*6};
OP=5.X: X*5 := {X*1, X*3, X*5};
OP=6.X: X*6 := {X*1, X*2, X*4};
];
```



```
X=X-S4: [ping(mark=3)*($-1).X-S2];
starBUG=7.X:
7.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*3, X*4, X*5};
OP=2.X: X*2 := {X*2, X*4, X*6, X*7};
OP=3.X: X*3 := {X*1, X*5, X*6, X*7};
OP=4.X: X*4 := {X*1, X*2, X*6, X*7};
OP=5.X: X*5 := {X*1, X*3, X*5};
OP=6.X: X*6 := {X*2, X*3, X*4};
OP=7.X: X*7 := {X*2, X*3, X*4, X*7};
];
```

iPOP*6.X-S1 = 6.X-S2; => $6.X-S1 = $6.X-S2;

Ist die Menge $7.X-S3 mit $7.X-S4 identisch? Wäre sehr schön. Gibt es einen Strukturerhaltungssatz bei bestimmten Permutationsoperationen (structurePOPs) von Noptronischen Mengen? Ist etwa ping so ein POP? Immerhin ergibt sich der POP ja aus der Struktur der Menge selbst. Oder sind etwa alle möglichen POPs sofern auch durchführbar structurePOPs? Nein, es gibt auch Operationen wie das Vertauschen bestimmter Elemente nur in den Definitionsmengen welche aus gleichen Mengenstrukturen unterschiedliche erzeugen. Hier agiert der POP unabhängig von der Struktur der Menge, gewissermaßen willkürlich. Das erklärt natürlich alles.

Einführung

Die mit dem Curch-Numeral k iterierte Funktion metaMAN addiert einfach die Mächtigkeit aller im Mengenterm nach der Darstellung von k.norm vorkommenden Elemente. k.norm ist eine rein formale Darstellung der Menge und verändert die Menge selbst nicht. Mit dieser veränderten Darstellung der Mengenterme wird jedoch eine selektive Anwendung von CARD mit nachfolgender Verarbeitung durch das miniPOOPI add von den k.norm bestimmten Elementen möglich.
Die Iteration der Mannigfaltigkeit erschafft Zahlenreihen deren Quotientenfolge vermuteterweise immer konvergiert. Dieser Grenzwert heißt Differentialquotient der Menge $.X und repräsentiert nach Ansicht namhafter Mathematiker wie z.B. Gritli Moser wichtige Eigenschaften jener Mengen.

```
DEF=norm:
[
type=norm:
var: k;
k.norm := col(k-1).norm;
];
```

Anwendungsbeispiele:
k=k: $.X:k.norm = colX*p:(k-1).norm;
k=0: $.X:0.norm = $.X;
k=1: $.X:1.norm = colX*p:0.norm = colX*p;
k=2: $.X:2.norm = colX*p:1.norm;

Hier konkret mal angewendet bei dem bekanten Sternwander X-Matrix mit $=3:
$3.X-Matrix:0.norm = $3.X-Matrix;
$3.X-Matrix:1.norm = {X*1:0.norm, X*2:0.norm, X*3:0.norm} = {X*1, X*2, X*3};
$3.X-Matrix:2.norm = {X*1:1.norm, X*2:1.norm, X*3:1.norm} =
= {{X*2, X*3}, {X*1, X*3}, {X*1, X*2}};
$3.X-Matrix:3.norm = {X*1:2.norm, X*2:2.norm, X*3:2.norm} =
= {{X*2:1.norm, X*3:1.norm}, {X*1:1.norm, X*3:1.norm}, {X*1:1.norm, X*2:1.norm}} =
= {{{X*1, X*3}, {X*1, X*2}}, {{X*2, X*3}, {X*1, X*2}}, {{X*2, X*3}, {X*1, X*3}}};

```
DEF=metaMAN:
[
nopFUN=metaMAN:
var: k;
k.metaMAN := addCARD:k.norm;
];
```

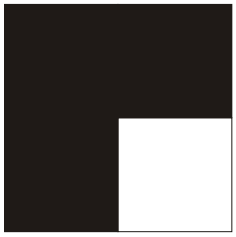
Hinweis zur Funktionsweise von addCARD: Das miniPOOPI add summiert die durch die Anwendung der Funktion CARD erzeugten Kardinalitäten der unter k.norm entstehenden Mengenterme.

Anwendung:
k=0: 0.metaMAN*set = addCARD*set:0.norm = CARD*set;
k=1: 1.metaMAN*set = addCARD*set:1.norm = MAN*set;
k=2: 2.metaMAN*set = addCARD*set:2.norm;
k=k: 3.metaMAN*set = addCARD*set:k.norm;

3.metaMAN*$3.X = addCARD*$3.X:3.norm =
addCARD*{{{X*1, X*3}, {X*1, X*2}}, {{X*2, X*3}, {X*1, X*2}}, {{X*2, X*3}, {X*1, X*3}}}
= 24;
Oder man läßt sich die Sternwandermenge unter k.norm im starBUG anzeigen um die iterierte Mannigfaltigkeit von Hand auszuzählen.

"Rechnen wir das Ding einfach mal am starBUG manuell durch, um zu sehen was da passiert!" säuselte die kluge Gritli-KI wieder mal so infantil kindlich zärtlich inszeniert, daß mir gleich Grooci&Vooci wurde. Ich öffnete meine letzte Dose Germania Ur-Pils und begann zu rechnen. Mmmmh, mal sehen, das sieht einfach aber verwickelt aus, diese unfundierten Mengensysteme stecken voller Überraschungen. Aaaalso, beginnen wir mit der Mächtigkeit der Matrixmenge:

```
X=X-omega1: [user];
starBUG=2.X-omega1:
2.[OP=index.X: X*p := X*p:+5.norm] =
[
OP=1.X:  X*1 := {X*1, X*2} =
{{X*1, X*2}, {X*1}} =
{{{X*1, X*2}, {X*1}}, {{X*1, X*2}}}=
{{{{X*1, X*2}, {X*1}}, {{X*1, X*2}}}, {{{X*1, X*2}, {X*1}}}} =
{{{{{X*1, X*2}, {X*1}}, {{X*1, X*2}}}, {{{X*1, X*2}, {X*1}}}},
{{{{X*1, X*2}, {X*1}}, {{X*1, X*2}}}}};

OP=2.X:  X*2 := {X*1} =
{{X*1, X*2}} =
{{{X*1, X*2}, {X*1}}}=
{{{{X*1, X*2}, {X*1}}, {{X*1, X*2}}}} =
{{{{{X*1, X*2}, {X*1}}, {{X*1, X*2}}}, {{{X*1, X*2}, {X*1}}}}};
];
=> $2.X-omega1 = {X*1, X*2};
```

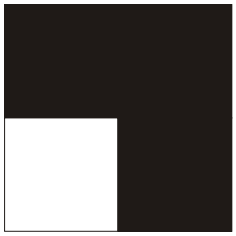


$2.X-omega1 hat genau 1 primär-reflexives Element.

Folge:
2, 2+1, 2+1+2, 2+1+2+2+1, 2+1+2+2+1+2+1+2, 2+1+2+2+1+2+1+2+2+1+2+2+1,
2+1+2+2+1+2+1+2+2+1+2+2+1+2+1+2+2+1+2+1+2 .... =
2, 3, 5, 8, 13, 21, 34..... = Fibonacci; Wie kommt der denn in ein unfundiertes Mengensystem?

Er wurde eingeschleppt durch Falschpark.
Beweis!

```
X=X-omega2: [user];
starBUG=2.X-omega2:
2.[OP=index.X: X*p := X*p:+4.norm] =
[
OP=1.X:  X*1 := {X*1, X*2} =
{{X*1, X*2}, {X*2}} =
{{{X*1, X*2}, {X*2}}, {{X*2}}} =
{{{{X*1, X*2}, {X*2}}, {{X*2}}}, {{{X*2}}}};

OP=2.X:  X*2 := {X*2} =
{{X*2}} =
{{{X*2}}}=
{{{{X*2}}}};
];
=> $2.X-omega2 = {X*1, X*2};
```



$.X-omega2 hat genau 2 primär-reflexive Elemente!

Folge: 2, 3, 4, 5, 6, 7, ... = ah ja, schönes Zählen!

Hinweis: Das Pluszeichen bei +k.norm bewirkt das Anzeigen aller Terme der Definitionsmengen von 1.norm bis k.norm, ansonsten würde nur der Term k.norm angezeigt. 1.norm = norm entspricht der Orthonormalform.

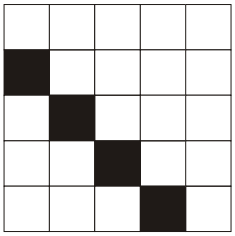
X=X-String;
starBUG=5.X-String:
5.[OP=index.X: X*p := X*p:+5.norm] =
[
OP=1.X: X*1 := { };

OP=2.X: X*2 := {X*1} =
{{ }};

OP=3.X: X*3 := {X*2} =
{{X*1}} =
{{{ }}};

OP=4.X: X*4 := {X*3} =
{{X*2}} =
{{{X*1}}} =
{{{{ }}}};

OP=5.X: X*5 := {X*4} =
{{X*3}} =
{{{X*2}}} =
{{{{X*1}}}} =
{{{{{ }}}}};
];
=> $5.X-String = {X*1, X*2, X*3, X*4, X*5}; Folge: 5, 4, 3, 2, 1, 0, 0, ... count down

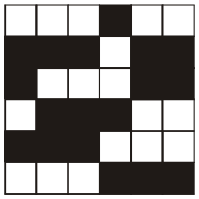


$5.X-String

Hier 2 komplementäre Mengen mit gleichem Nopos und auch gleichen(?) Folgen:

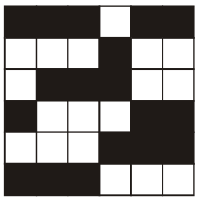
X := [user];

starBUG=6.X:
6.[OP=index.X: X*p := X*p:+2.norm] =
[
OP=1.X: X*1 := {X*4} =
{{X*2, X*3, X*4}};
OP=2.X: X*2 := {X*1, X*2, X*3, X*5, X*6} =
{{X*4}, {{X*1, X*2, X*3, X*5, X*6}}, {{X*1, X*5, X*6}}, {{X*1, X*2, X*3}}, {{X*4, X*5, X*6}}};
OP=3.X: X*3 := {X*1, X*5, X*6} =
{{X*4}, {{X*1, X*2, X*3}}, {{X*4, X*5, X*6}}};
OP=4.X: X*4 := {X*2, X*3, X*4} =
{{{X*1, X*2, X*3, X*5, X*6}}, {{X*1, X*5, X*6}}, {{X*2, X*3, X*4}}};
OP=5.X: X*5 := {X*1, X*2, X*3} =
{{X*4}, {{X*1, X*2, X*3, X*5, X*6}}, {{X*1, X*5, X*6}}};
OP=6.X: X*6 := {X*4, X*5, X*6} =
{{{X*2, X*3, X*4}}, {{X*1, X*2, X*3}}, {{X*4, X*5, X*6}}};
];
Folge: 6, 18, 54, ...



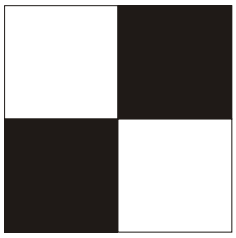
$6.X

starBUG=6.X':
6.[OP=index.X: X*p := X*p:+2.norm] =
[
OP=1.X: X*1 := {X*1, X*2, X*3, X*5, X*6} =
{{{X*1, X*2, X*3, X*5, X*6}}, {X*4}, {{X*2, X*3, X*4}}, {{X*4, X*5, X*6}}, {{X*1, X*2, X*3}}};
OP=2.X: X*2 := {X*4} =
{{X*1, X*5, X*6}};
OP=3.X: X*3 := {X*2, X*3, X*4} =
{{{X*4}}, {{X*2, X*3, X*4}}, {{X*1, X*5, X*6}}};
OP=4.X: X*4 := {X*1, X*5, X*6} =
{{{X*1, X*2, X*3, X*5, X*6}}, {{X*4, X*5, X*6}}, {{X*1, X*2, X*3}}};
OP=5.X: X*5 := {X*4, X*5, X*6} =
{{{X*1, X*5, X*6}}, {{X*4, X*5, X*6}}, {{X*1, X*2, X*3}}};
OP=6.X: X*6 := {X*1, X*2, X*3} =
{{{X*1, X*2, X*3, X*5, X*6}}, {{X*4}}, {{X*2, X*3, X*4}}};
];
Folge: 6, 18, 54, ...



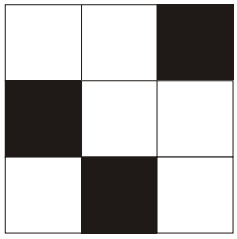
$6.X'

```
X=X-Ring;
starBUG=2.X-Ring:
2.[OP=index.X: X*p := X*p:+5.norm] =
[
OP=1.X:  X*1 := {X*2} =
{{X*2}} =
{{{X*1}}} =
{{{{X*2}}}},
{{{{{X*2}}}}};

OP=2.X:  X*2 := {X*1} =
{{X*2}} =
{{{X*1}}} =
{{{{X*2}}}},
{{{{{X*1}}}}};
];
=> $2.X-Ring = {X*1, X*2}; Folge: 2, 2, 2, 2, 2, 2, 2, .....
```



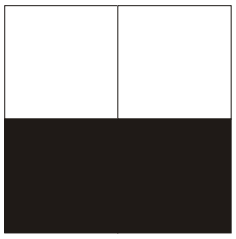
$2.X-Ring

```
X=X-Ring;
starBUG=3.X-Ring:
3.[OP=index.X: X*p := X*p:+5.norm] =
[
OP=1.X:  X*1 := {X*3} =
{{X*2}} =
{{{X*1}}} =
{{{{X*3}}}},
{{{{{X*2}}}}};

OP=2.X:  X*2 := {X*1} =
{{X*3}} =
{{{X*1}}} =
{{{{X*2}}}},
{{{{{X*1}}}}};

OP=3.X: X*3 := {X*2} =
{{X*1}} =
{{{X*3}}} =
{{{{X*2}}}},
{{{{{X*1}}}}};
];
=> $3.X-Ring = {X*1, X*2, X*3}; Folge: 3, 3, 3, 3, 3, 3, 3, .....
```



$3.X-Ring

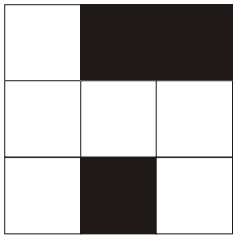
```
X=X-omega3;
starBUG=2.X-omega3:
2.[OP=index.X: X*p := X*p:+3.norm] =
[
OP=1.X: X*1 := { };

OP=2.X: X*2 := {X*1, X*2}
= {{ }, {X*1, X*2}} =
{{ }, {{ }, {X*1, X*2}}};
];
=> $2.X-omega3 = {X*1, X*2}; Folge: 2, 2, 2, 2, 2, 2, 2, ......
```



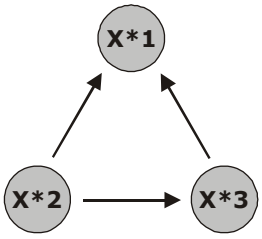
$2.X-omega3

```
starBUG=3.X:
3.[OP=index.X: X*p := X*p:+3.norm] =
[
OP=1.X:  X*1 := {X*2, X*3,} =
{{ }, {X*2}} =
{{ }, {{ }}};

OP=2.X:  X*2 := { };

OP=3.X:  X*3 := {X*2} =
{{ }};
];
=> $3.X = {X*1, X*2, X*3};
```

Folge: 3, 3, 1, 0, 0, ...



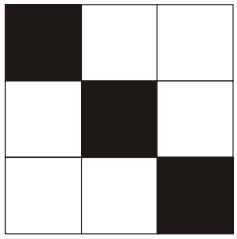
$3.X



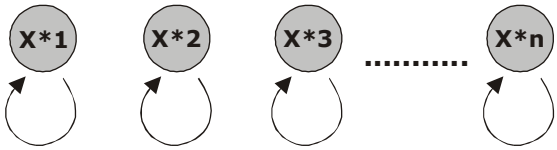
Reale Sternwandermengen kommen wohl immer zu einem Stillstand. Da strandet man immer im "Nichts". Das entspricht in der Tat den ordentlichen Turing-Maschinen. Sie sollen ja auch mal fertig werden damit man ein Ergebnis hat. Operative Abgeschlossenheit ist eben sehr beliebt. Da weiß man was man hat.

```
X=X-StarDust;
starBUG=3.X-StarDust:
3.[OP=index.X: X*p := X*p:+5.norm] =
[
OP=1.X:  X*1 := {X*1} =
{{X*1}} =
{{{X*1}}} =
{{{{X*1}}}},
{{{{{X*1}}}}};

OP=2.X:  X*2 := {X*2} =
{{X*2}} =
{{{X*2}}} =
{{{{X*2}}}},
{{{{{X*2}}}}};

OP=3.X: X*3 := {X*3} =
{{X*3}} =
{{{X*3}}} =
{{{{X*3}}}},
{{{{{X*3}}}}};
];
=> $3.X-StarDust = {X*1, X*2, X*3};
```

Folge: 3, 3, 3, 3, 3, 3, 3, ...



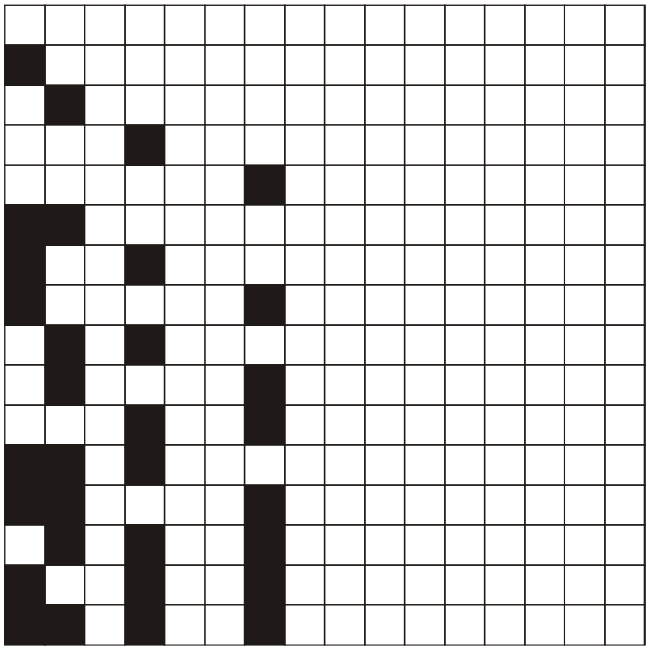
$3.X-StarDust



Der Omegaspace $n.X-StarDust, unter Astrophysikern auch bekannt als die mysteriöse Vakuumenergie. Oft kopiert, aber unerreicht. Perfekte Symmetrie in totaler Zersplitterung, der absolute Stillstand auf höchstem Niveau.

X-omsetBIGBANG := [$1.X := omset;];

$1.X = omset = {omset};
$2.X = {{ }, omset};
$3.X ={{ }, {{ }}, omset, {{ }, omset}};
$4.X =
{
{ },
{{ }}, {{{ }}}, omset, {{{ }, omset}},
{{ }, {{ }}},
{{ }, omset},
{{ }, {{ }, omset}},
{{{ }}, omset},
{{{ }}, {{ }, omset}},
{omset, {{ }, omset}},
{{ }, {{ }}, omset},
{{ }, {{ }}, {{ }, omset}},
{{{ }}, omset, {{ }, omset}},
{{ }, omset, {{ }, omset}},
{{ }, {{ }}, omset, {{ }, omset}}
};



$4.X-omsetBIGBANG

starBUG=4.omsetBIGBANG:
16.[OP=index.X: X*p := X*P:norm+] =
[
OP=01.X: X*01 := { };
OP=02.X: X*02 := {{ }} = {X*01} = {{ }};
OP=03.X: X*03 := {{{ }}} = {X*02} = {{X*01}} = {{{ }}};
OP=04.X: X*04 := omset = {X*04} = {{X*04}} = {{{X*04}}};
OP=05.X: X*05 := {{{ }, omset }} = {X*07} ={X*01, X*04} = {{ }, {X*04}};
OP=06.X: X*06 := {{ }, {{ }}} = {X*01, X*02} = {{ }, {X*01}} = {{ }, {{ }}};
OP=07.X: X*07 := {{ }, omset }= {X*01, X*04} = {{ }, {X*04}} = {{ }, {{X*04}}};
OP=08.X: X*08 := {{ }, {{ }, omset}} = {X*01, X*07} = {{ }, {X*01, X*04}} =
{{ }, {{ }, {X*04}}};
OP=09.X: X*09 := {{{ }}, omset} = {X*02, X*04} = {{X*01}, {X*04}} =
{{{ }}, {X*04}};
OP=10.X: X*10 := {{{ }}, {{ }, omset}} = {X*02, X*07} = {{X*01}, {X*01, X*04}} =
{{{ }}, {{ }, {X*04}}};
OP=11.X: X*11 := {omset, {{ }, omset}} = {X*04, X*07} = {{X*04}, {X*01, X*04}} =
{{{X*04}}, {{ }, {X*04}}};
OP=12.X: X*12 := {{ }, {{ }}, omset} = {X*01, X*02, X*04} = {{ }, {X*01}, {X*04}} =
{{ }, {{ }}, {X*04}};
OP=13.X: X*13 := {{ }, {{ }}, {{ }, omset}} = {X*01, X*02, X*07} =
{{ }, {X*01}, {X*01, X*04}} = {{ }, {{ }}, {{ }, {X*04}}};
OP=14.X: X*14 := {{{ }}, omset, {{ }, omset}} = {X*02, X*04, X*07} =
{{X*01}, {X*04}, {X*01, X*04}} = {{{ }}, {{X*04}}, {{ }, {X*04}}};
OP=15.X: X*15 := {{ }, omset, {{ }, omset}} = {X*01, X*04, X*07} =
{{ }, {X*04}, {X*01, X*04}} = {{ }, {{X*04}}, {{ }, {{X*04}}}};
OP=16.X: X*16 := {{ }, {{ }}, omset, {{ }, omset}} = {X*01, X*02, X*04, X*07} =
{{ }, {X*01}, {X*04}, {X*01, X*04}} = {{ }, {{ }}, {{X*04}}, {{ }, {X*04}}};
]

Folge: 16, 32, 32, 16, 16, 16, ...

Der Potenzmengen-Sternwander mit omset als Startelement scheint nicht so interessant zu sein. Das Omega-Element bleibt am Schluß übrig und erzeugt so eine konstante Folge. Da kann man nix machen. Am besten erst mal ein Butterbrot essen. Danach ganz langsam abziehen und mit viel Weizenbier ablöschen.

starBUG=7.X-Space:
7.[OP=index.X: X*p := X*p:+6.norm] =
[
OP=1.X: X*1 := { };

OP=2.X: X*2 := {X*1} =
{{ }};

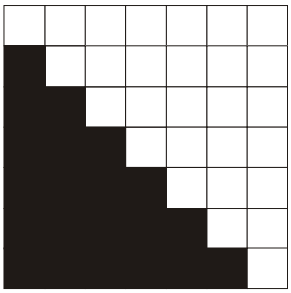
OP=3.X: X*3 := {X*1, X*2} =
{{ }, {X*1}} =
{{ }, {{ }}};

OP=4.X: X*4 := {X*1, X*2, X*3} =
{{ }, {X*1}, {X*1, X*2}} =
{{ }, {{ }}, {{ }, {X*1}}} =
{{ }, {{ }}, {{ }, {{ }}}};

OP=5.X: X*5 := {X*1, X*2, X*3, X*4} =
{{ }, {X*1}, {X*1, X*2}, {X*1, X*2, X*3}} =
{{ }, {{ }}, {{ }, {X*1}}, {{ }, {X*1}, {X*1, X*2}}} =
{{ }, {{ }}, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {X*1}}}} =
{{ }, {{ }}, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {{ }}}}};

OP=6.X: X*6 := {X*1, X*2, X*3, X*4, X*5} =
{{ }, {X*1}, {X*1, X*2}, {X*1, X*2, X*3}, {X*1, X*2, X*3, X*4}} =
{{ }, {{ }}, {{ }, {X*1}}, {{ }, {X*1}, {X*1, X*2}}, {{ }, {X*1}, {X*1, X*2}, {X*1, X*2, X*3}}} =
{{ }, {{ }}, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {X*1}}}, {{ }, {{ }}, {{ }, {X*1}}, {{ }, {X*1}, {X*1, X*2}}}} =
{{ }, {{ }}, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {{ }}}}, {{ }, {{ }}, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {X*1}}}}} =
{{ }, {{ }}, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {{ }}}}, {{ }, {{ }}, {{ }, {{ }}}, {{ }, {{ }}, {{ }, {{ }}}}}};

OP=7.X: X*7 := {X*1, X*2, X*3, X*4, X*5, X*6} =
{
{ },
{X*1},
{X*1, X*2},
{X*1, X*2, X*3},
{X*1, X*2, X*3, X*4},
{X*1, X*2, X*3, X*4, X*5}
} =
{
{ },
{{ }},
{{ }, {X*1}},
{{ }, {X*1}, {X*1, X*2}},
{{ }, {X*1}, {X*1, X*2}, {X*1, X*2, X*3}},
{
{ }, {{ }}, {{ }, {X*1}}, {{ }, {X*1}, {X*1, X*2}}, {{ }, {X*1}, {X*1, X*2},
{X*1, X*2, X*3}
}
} = [ERROR=insufficient_wrtln_for_X*p:+4.norm];
];



$7.X-Space

=> $7.X-Space = {X*1, X*2, X*3, X*4, X*5, X*6, X*7};

Eine Auszählung der Elemente im starBUG ergibt die Folge: 7, 21, 35, 35, 21, 7, 1, 0, 0, ...
Allgemeiner Ansatz wenn n := CARD*$n.X-Space;

k=0: addCARD*$n.X-Space:0.norm=CARD*$.X-Space=(n-0)/1!;
k=1: metaMAN*$n.X-Space:0.norm=addCARD*$n.X-Space:1.norm=(n-0)*(n-1)/2!;
k=2: metaMAN*$n.X-Space:1.norm=addCARD*$n.X-Space:2.norm=(n-0)*(n-1)*(n-2)/3!;
.....
k=k: metaMAN*$n.X-Space:(k-1).norm=addCARD*$n.X-Space:k.norm=(n-0)**(n-k)/(k+1)!;
(Ha, ha! Jetzt hat der Depp auch noch den Binomialkoeffizienten neu entdeckt! Anmerkung der Redaktion)

Als handlicher Zahlenblock sieht das dann so aus:
Jede Zeile stellt die iterierte Mannigfaltigkeit als Zahlenfolge einer Gradation von $.X-Space dar.
Jede Spalte entspricht k beginnend bei k=0;

OP=k: 001, 002, 003, 004, 005, 006, 007, 008, 009, 010, 011, 012, 013, 014, 015, 0.ALEPH;
$=01: 001, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 0.ALEPH;
$=02: 002, 001, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 0.ALEPH;
$=03: 003, 003, 001, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 0.ALEPH;
$=04: 004, 006, 004, 001, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 0.ALEPH;
$=05: 005, 010, 010, 005, 001, 000, 000, 000, 000, 000, 000, 000, 000, 000, 000, 0.ALEPH;
$=06: 006, 015, 020, 015, 006, 001, 000, 000, 000, 000, 000, 000, 000, 000, 000, 0.ALEPH;
$=07: 007, 021, 035, 035, 021, 007, 001, 000, 000, 000, 000, 000, 000, 000, 000, 0.ALEPH;
$=08: 008, 028, 056, 070, 056, 028, 008, 001, 000, 000, 000, 000, 000, 000, 000, 0.ALEPH;
$=09: 009, 036, 084, 126, 126, 084, 036, 009, 001, 000, 000, 000, 000, 000, 000, 0.ALEPH;
$=10: 010, 045, 120, 210, 252, 210, 120, 045, 010, 001, 000, 000, 000, 000, 000, 0.ALEPH;
$=11: 011, 055, 165, 330, 462, 462, 330, 165, 055, 011, 001, 000, 000, 000, 000, 0.ALEPH;
$=12: 012, 066, 220, 495, 792, 924, 792, 495, 220, 066, 012, 001, 000, 000, 000, 0.ALEPH;
$=0.ALEPH;

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1
1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1
1 14 91 364 1001 2002 3003 3432 3003 2002 1001 364 91 14 1
1 15 105 455 1365 3003 5005 6435 6435 5005 3003 1365 455 105 15 1
1 16 120 560 1820 4368 8008 11440 12870 11440 8008 4368 1820 560 120 16 1

Selbstportrait nach Sauna

Links sehen sie den Blaise Pascal kurz nach der Erfindung seines Dreiecks. Angeblich wurde er dazu von hüpfenden Filzläusen im Schamdreieck einer Yoga-Prinzessin in der Sauna inspiriert.
Rechts sieht man den Erfinder von $.X-Space, John von Neumann, vor dem Magischen Zentralcomputer von Cocos*Palam. Sehr wahrscheinlich handelt es sich dabei noch um den MAC 99/111 mit Borland-Compiler und einem rein mechanischen Prozessor. Die Programmierung erfolgte über Steckverbindungen in Form von 6,3 mm Klinkensteckern in Stereo.

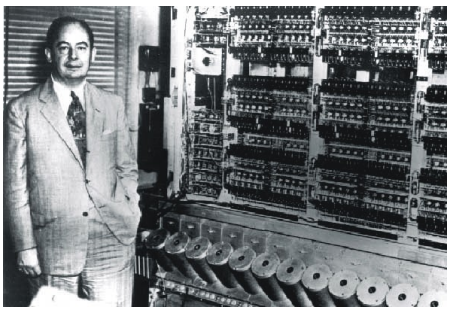
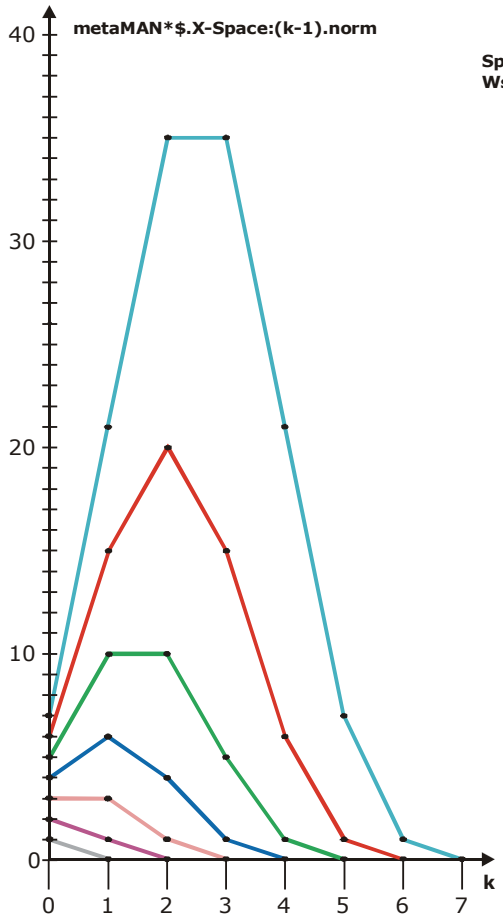
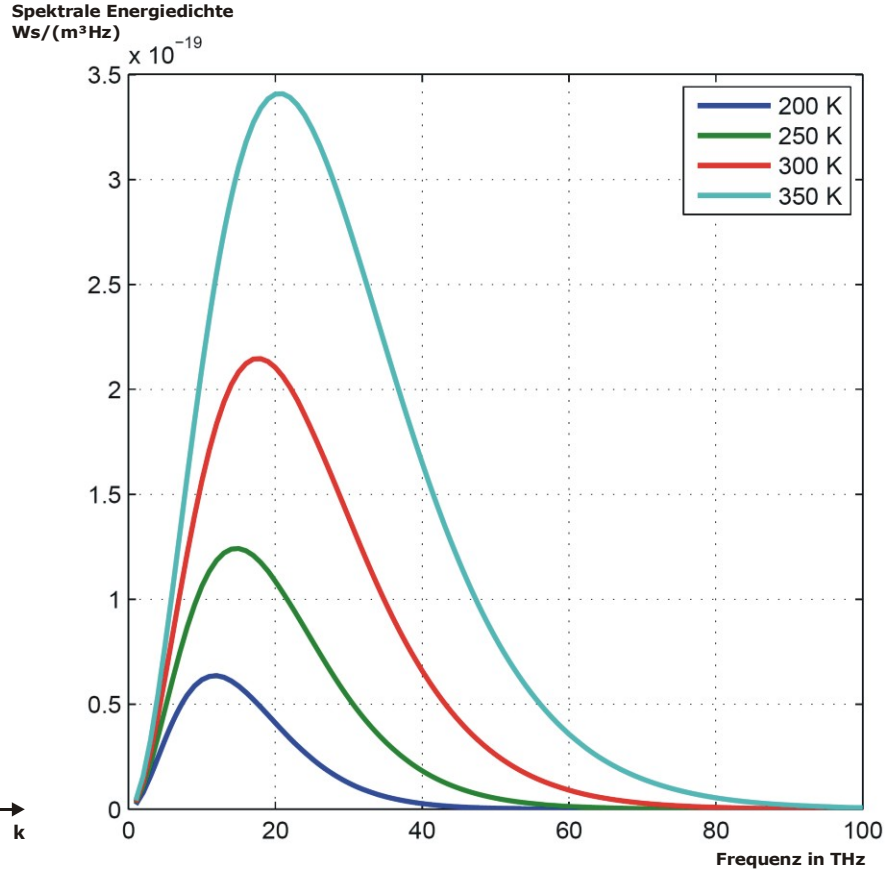


Photo by Lexee Smith/1944

Wir freuen uns über so viel schöne Zusammenhänge und essen erst mal ein Butterbrot. Weder der Blaise noch der John konnten ahnen, daß ein Etwas sie miteinander verbindet. Wie es ja in der Dementia Simplex schon prophezeit wurde in Regel 4: Verbinde alle Verbindungen mit einer Binde. Am Ende sieht man das Universum als Mumie... Aber es kommt noch besser!





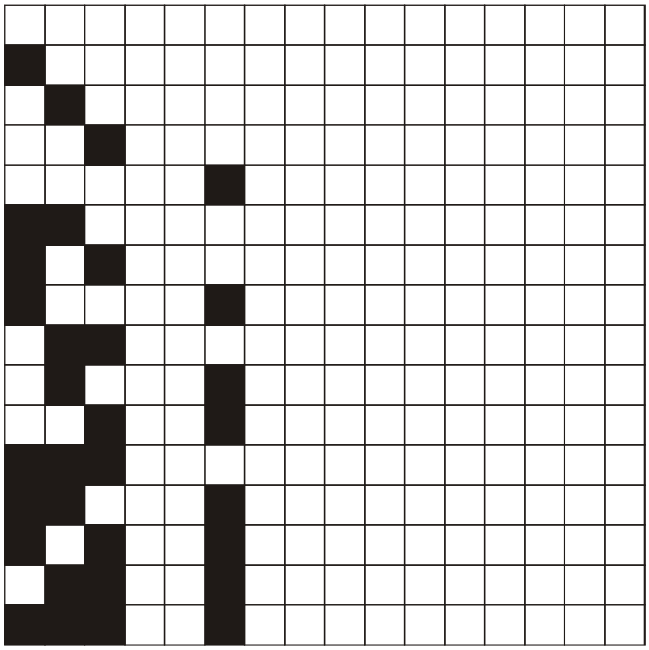
metaMAN*$7.X-Space:(k-1).norm
metaMAN*$6.X-Space:(k-1).norm
metaMAN*$5.X-Space:(k-1).norm
metaMAN*$4.X-Space:(k-1).norm
metaMAN*$3.X-Space:(k-1).norm
metaMAN*$2.X-Space:(k-1).norm
metaMAN*$1.X-Space:(k-1).norm

Planck-Spektrum bei verschiedenen Temperaturen
Messungen und Grafik by Gritli Moser/Quantum Lab TNSC/Germany

Hat das Planck-Spektrum etwas mit dem Pascal-Dreieck und den von Neumann-Zahlen zu tun? Und wenn ja, was genau? Wenn man den Wert von k in der linken Grafik weiter erhöht und entsprechend unproportional skaliert, wird ein immer runderer erscheinender Graph erzeugt, der dem Graphen des Planck-Spektrums sich bei stetig höher werdenden Temperaturen beliebig annähert. Das behauptet jedenfalls die Gritli.
Gritli, eines Tages bekommst du doch noch die Fields-Medaille oder den Nobel-Preis für erfundene Wissenschaften. Hast du sonst noch irgendwelche Theorien am Start?

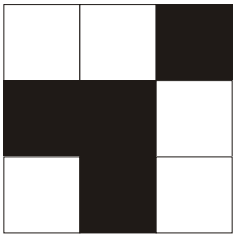
Eijooo! Paß auf, paß auf! Kommt eine Blondine, ein Neger und ein Jude zum Arzt. Sagt die Blondine der Neger ist ja gar kein Neger. Sagt der Neger die Blondine ist ja auch gefärbt. Die Blondine darauf kuckt arrogant und beleidigt nach oben. Mischt sich der Jude ein. Ich bin zwar Jude, aber nicht farbenblind. Der Neger ist blond und die Blondine eher braun. Sagt der Neger, er sei Long Dong Silver und damit wie geschaffen für die Blondine. Mischt sich jetzt der Arzt ein, ein blonder jüdisch stämmiger Nigerianer und fragt die Blondine, ob sie einen Sehtest machen wolle, denn sie sei ja wohl voll farbenblind und auch antisemitisch. Der Neger unterdessen packt eine Tafel Sarotti-Schokolade aus und streckt dem Juden die Zunge raus. Daraufhin lacht der Arzt sich fast ohnmächtig und zeigt der Blondine den Stinkfinger. Jetzt reißen sich alle die Klamotten von den erhitzten Leibchen und der Jude beginnt zu beten. Der Neger holt unterdessen seine Steppschuhe aus einer lila Felltasche steppt und singt "Singing in the Rain". Der Jude schlägt nun der antisemitischen Blondine die Kippa runter und darunter kommt ein nackt rasiertes Rentier mit Hakenkreuztattoos hervor. Alle lachen und fallen sich sabbernd in die Arme. Danach bestellen sie ein Zigeunerschnitzel beim Türken um die Ecke und bezahlen in Reichsmark. Mensch Gritli, dein Humor ist ja sehr komplex. Ich liebe Witze erzählende Turing-Maschinen. Das war spitze!

starBUG=4.X-BIGBANG:
16.[OP=index.X: X*p := X*p:+3.norm] =
[
OP=01.X: X*01 := { };
OP=02.X: X*02 := {X*1} =
{{ }};
OP=03.X: X*03 := {X*2} =
{{X*1}} =
{{{ }}};
OP=04.X: X*04 := {X*3} =
{{X*2}} =
{{{X*1}}};
OP=05.X: X*05 := {X*6} =
{{X*1, X*2}} =
{{{ }, {X*1}}};
OP=06.X: X*06 := {X*1, X*2} =
{{ }, {X*1}} =
{{ }, {{ }};
OP=07.X: X*07 := {X*1, X*3} =
{{ }, {X*2}} =
{{ }, {{X*1}}};
OP=08.X: X*08 := {X*1, X*6} =
{{ }, {X*1, X*2}} =
{{ }, {{ }, {X*1}}};
OP=09.X: X*09 := {X*2, X*3} =
{{X*1}, {X*2}} =
{{{ }}, {{X*1}}};
OP=10.X: X*10 := {X*2, X*6} =
{{X*1}, {X*1, X*2}} =
{{{ }}, {{ }, {X*1}}};
OP=11.X: X*11 := {X*3, X*6} =
{{X*2}, {X*1, X*2}} =
{{{X*1}}, {{ }, {X*1}}};
OP=12.X: X*12 := {X*1, X*2, X*3} =
{{ }, {X*1}, {X*2}} =
{{ }, {{ }}, {{X*1}}};
OP=13.X: X*13 := {X*1, X*2, X*6} =
{{ }, {X*1}, {X*1, X*2}} =
{{ }, {{ }}, {{ }, {X*1}}};
OP=14.X: X*14 := {X*1, X*3, X*6} =
{{ }, {X*2}, {X*1, X*2}} =
{{ }, {{X*1}}, {{ }, {X*1}}};
OP=15.X: X*15 := {X*2, X*3, X*6} =
{{X*1}, {X*2}, {X*1, X*2}} =
{{{ }}, {{X*1}}, {{ }, {X*1}}};
OP=16.X: X*16 := {X*1, X*2, X*3, X*6} =
{{ }, {X*1}, {X*2}, {X*1, X*2}} =
{{ }, {{ }}, {{X*1}}, {{ }, {X*1}}};
];
=>
$4.X-BIGBANG =
{X*01, X*02, X*03, X*04, X*05, X*06, X*07, X*08,
X*09, X*10, X*11, X*12, X*13, X*14, X*15, X*16};



$4.X-BIGBANG

Folge: 16, 1+1+1+1+2+2+2+2+2+2+3+3+3+3+4, 1+1+2+1+1+2+2+3+3+2+3+3+4+4,
1+1+1+1+1+1+2+1+1+2+2+2, 0, 0, ... =
16, 32, 32, 16, 0, 0, ... = 2 hoch 4, 2 hoch 5, 2 hoch 5, 2 hoch 4, 0, 0, ...



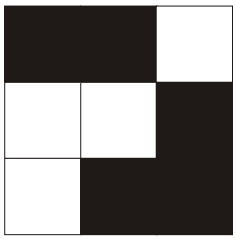
$3.X-T1

```
X=X-T1: [user];
starBUG=3.X-T1:
3.[OP=index.X: X*p := X*p:+5.norm] =
[
OP=1.X: X*1 := {X*3} =
{{X*2}} =
{{{X*1, X*2}}} =
{{{{X*3}, {X*1, X*2}}}} =
{{{{{X*2}}, {{X*3}, {X*1, X*2}}}}};

OP=2.X: X*2 := {X*1, X*2} =
{{X*3 }, {X*1, X*2}} =
{{{X*2}}, {{X*3}, {X*1, X*2}}} =
{{{{X*1, X*2}}}, {{{X*2}}, {{X*3}, {X*1, X*2}}}} =
{{{{{X*3}, {X*1, X*2}}}}, {{{{X*1, X*2}}}, {{{X*2}}, {{X*3}, {X*1, X*2}}}}};

OP=3.X: X*3 := {X*2} =
{{X*1, X*2}}=
{{{X*3}, {X*1, X*2}}} =
{{{{X*2}}, {{X*3}, {X*1, X*2}}}} =
{{{{{X*1, X*2}}}, {{{X*2}}, {{X*3}, {X*1, X*2}}}}};
];
=>
$3.X-T1 = {X*1, X*2, X*3};
```

Folge: 3, 4, 6, 9, 13, 19, 28, 41, 60, 88, 129, 189, 277, 406, 595, 872, 1278, 1873, 2745, 4023, 5896, 8641, 12664, 18560, 27201, 39865, 58425, 85626, 125491, 183916, ...

3+6=9; 4+9=13; 6+13=19; 9+19=28; 13+28=41; ...= Fibonacci Flip oder Fliponacci oder auch offiziell in OEIS unter A000930 registriert als Narayana's cows sequence:
a(0) := a(1) := a(2) := 1; a(n) := a(n-1) + a(n-3);

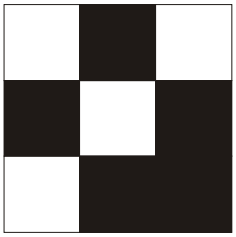
Die Quotientenfolge scheint auch zu konvergieren wie beim Standard-Fibonacci.
Vermuteter Grenzwert hier: 1,465...



$3.X-T2

```
X=X-T2: [user];
starBUG=3.X-T2:
3.[OP=index.X: X*p := X*p:+4.norm] =
[
OP=1.X: X*1 := {X*1, X*2} =
{{X*1, X*2}, {X*3}} =
{{{X*1, X*2}, {X*3}}, {{X*2, X*3}}} =
{{{{X*1, X*2}, {X*3}}, {{X*2, X*3}}}, {{{X*3}, {X*2, X*3}}}};

OP=2.X: X*2 := {X*3} =
{{X*2, X*3}} =
{{{X*3}, {X*2, X*3}}} =
{{{{X*2, X*3}}, {{X*3}, {X*2, X*3}}}};

OP=3.X: X*3 := {X*2, X*3} =
{{X*3}, {X*2, X*3}} =
{{{X*2, X*3}}, {{X*3}, {X*2, X*3}}} =
{{{{X*3}, {X*2, X*3}}}, {{{X*2, X*3}}, {{X*3}, {X*2, X*3}}}};
];
=> $3.X-T2 = {X*1, X*2, X*3};
```

Folge: 3, 5, 8, 13, 21, ... = Verspäteter Fibonacci.



$3.X-ping

X=X-ping;
starBUG=3.X-ping:
3.[OP=index.X: X*p := X*p:+5.norm] =
[
OP=1.X: X*1 := {X*2} =
{{X*1, X*3}} =
{{{X*2}, {X*2, X*3}}} =
{{{{X*1, X*3}}, {{X*1, X*3}, {X*2, X*3}}}} =
{{{{{X*2}, {X*2, X*3}}}, {{{X*2}, {X*2, X*3}}, {{X*1, X*3}, {X*2, X*3}}}}};

OP=2.X: X*2 := {X*1, X*3} =
{{X*2}, {X*2, X*3}}=
{{{X*1, X*3}}, {{X*1, X*3}, {X*2, X*3}}} =
{{{{X*2}, {X*2, X*3}}}, {{{X*2}, {X*2, X*3}}, {{X*1, X*3}, {X*2, X*3}}}} =
{{{{{X*1, X*3}}, {{X*1, X*3}, {X*2, X*3}}}}, {{{{X*1, X*3}}, {{X*1, X*3}, {X*2, X*3}}},
{{{X*2}, {X*2, X*3}}, {{X*1, X*3}, {X*2, X*3}}}}};

OP=3.X: X*3 := {X*2, X*3} =
{{X*1, X*3}, {X*2, X*3}}=
{{{X*2}, {X*2, X*3}}, {{X*1, X*3}, {X*2, X*3}}} =
{{{{X*1, X*3}}, {{X*1, X*3}, {X*2, X*3}}}, {{{X*2}, {X*2, X*3}}, {{X*1, X*3},
{X*2, X*3}}}} =
{{{{{X*2}, {X*2, X*3}}}, {{{X*2}, {X*2, X*3}}, {{X*1, X*3}, {X*2, X*3}}}}, {{{{X*1,
X*3}}, {{X*1, X*3}, {X*2, X*3}}}, {{{X*2}, {X*2, X*3}}, {{X*1, X*3}, {X*2, X*3}}}}};
];
=>
$3.X-ping = {X*1, X*2, X*3};

Folge: 3, 5, 9+, 16-, 29+, 52-, 94+, 169-, 304+?, 546-?, 981+?, 1762-?, 3165+?, 5685-?, 10212+?, 18343-?, 32949+?, 59185-?, 106312+?, 190964-?, ...
Die Quotientenfolge scheint wohl zu konvergieren.
Vermuteter Grenzwert: 1,79626...

Teilbildungsgesetz der Folge:
metaMAN*$3.X:(k+1).norm := metaMAN*$3.X:k.norm + metaMAN*$3.X*3:k.norm

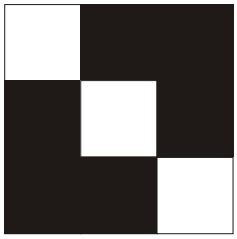
k=0;
metaMAN*$3.X:1.norm = metaMAN*$3.X:0.norm + metaMAN*$3.X*3:0.norm;
Probe: 9 = 5 + 4;
k=1;
metaMAN*$3.X:2.norm = metaMAN*$3.X:1.norm + metaMAN*$3.X*3:1.norm;
Probe: 16 = 9 + 7;
k=2;
metaMAN*$3.X:3.norm = metaMAN*$3.X:2.norm + metaMAN*$3.X*3:2.norm;
Probe: 29 = 16 + 13;
k=3;
metaMAN*$3.X:4.norm = metaMAN*$3.X:3.norm + metaMAN*$3.X*3:3.norm;
Probe: 52 = 29 + 23;
Geht doch auch so. Aber wie bastelt man eine Funktion, die immer das k-te Glied erzeugt?

Sei F(n) die Fibonacci-Folge dann produziert $3.X'-ping die Folge a(n) = F(n)+1;

a(n): (1, 2, 2,) 3, 4, 6, 9, 14, 22, 35, 56, ...
F(n): (0, 1, 1,) 2, 3, 5, 8, 13, 21, 34, 55, ...
3+4-1=6; 4+6-1=9; 6+9-1=14; 9+14-1=22; 14+22-1=35; 22+35-1=56; 35+56-1=90;
a(n) := a(n-1)+a(n-2)-1 = F(n)+1; a(0) := 1; a(1) := 2; F(0) := 0; F(1) := 1;

Frage: Wie verhalten sich Folgen von Komplementär-Sternwandern zahlentheoretisch zueinander?

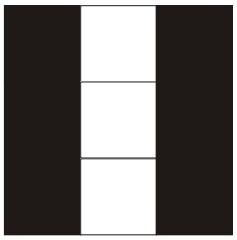
```
X=X-Matrix;
starBUG=3.X-Matrix:
3.[OP=index.X: X*p := X*p:+3.norm] =
[
OP=1.X:  X*1 := {X*2, X*3} =
{{X*1, X*3}, {X*1, X*2}} =
{{{X*2, X*3}, {X*1, X*2}}, {{X*2, X*3}, {X*1, X*3}}};

OP=2.X:  X*2 := {X*1, X*3} =
{{X*2, X*3}, {X*1, X*2}} =
{{{X*1, X*3}, {X*1, X*2}}, {{X*2, X*3}, {X*1, X*3}}};

OP=3.X: X*3 := {X*1, X*2} =
{{X*2, X*3}, {X*1, X*3}} =
{{{X*1, X*3}, {X*1, X*2}}, {{X*2, X*3}, {X*1, X*2}}};
];
=>
$3.X-Matrix = {X*1, X*2, X*3};
```



$3.X-Matrix

Folge: 3, 3*2, 3*4, 3*8, 3*16, 3*32, ...3*(2 hoch k); bei k.norm im starBUG
Allgemein: F(k) = |$.X|*(k.|X*p|*1);

```
X=X-fake: [user];
starBUG=3.X-fake:
3.[OP=index.X: X*p := X*p:+2.norm] =
[
OP=1.X: X*1 := {X*1, X*3} =
{{X*1, X*3}, {X*1, X*3}};

OP=2.X: X*2 := {X*1, X*3} =
{{X*1, X*3}, {X*1, X*3}};

OP=3.X: X*3 := {X*1, X*3} =
{{X*1, X*3}, {X*1, X*3}};
];
=> ERROR="totalMQL";
```



$3.X-fake

Folge: 3, 6, 12, 24, 48, 96, ... =3*(2 hoch k);
oder ganz allgemein: F(k) = |$.X|*(k.|X*p|*1);

Und das immer für alle Sternwandermengen deren Elemente alle identisch sind!
Klingt doch eher langweilig. Damit lassen sich alle Meta-Mannigfaltigkeitsreihen aller Fake-Sternwander beschreiben. So einfach sieht das bei den echten Sternwandern nicht aus. Hier gilt dieser Zusammenhang nur für Sternwandermengen deren Elemente alle die gleiche Mächtigkeit besitzen. Ist wohl eine Argument für die Wirklichkeit der Struktur der Sternwandermengen und für das Matrixmengenpostulat.
Vermutung: Für alle Mannigfaltigkeits-Folgen Noptronischer Mengen die nicht kollabieren, also keine Nullfolge erzeugen, nähert sich der Quotient zweier aufeinander folgender Zahlen einem Grenzwert. Beim Fibonacci kennt man den exakten Wert: (1 + sqrt*5)/2 Beim X-ping scheint sich ein Wert so um die 1,796... zu bilden und beim Fliponacci so um 1,465... Fragt sich halt wie diese Werte arithmetisch genau formuliert aussehen oder anders gesagt, repräsentieren diese noptronischen Mengen immer auch eine arithmetisch formulierbare Funktion? Können noptronische Mengen vielleicht sogar Zahlenfolgen erzeugen, die arithmetisch gar nicht darstellbar sind? Vielleicht sogar die Primzahlenfolge? Sind noptronische Mengen mächtiger als die Arithmetik? Denn wenn eine so einfache Menge wie $2.X-omega1 eine arithmetische Struktur wie die Formel von Binet hervorbringt, was für Dinger produzieren dann komplexere noptronische Mengen?

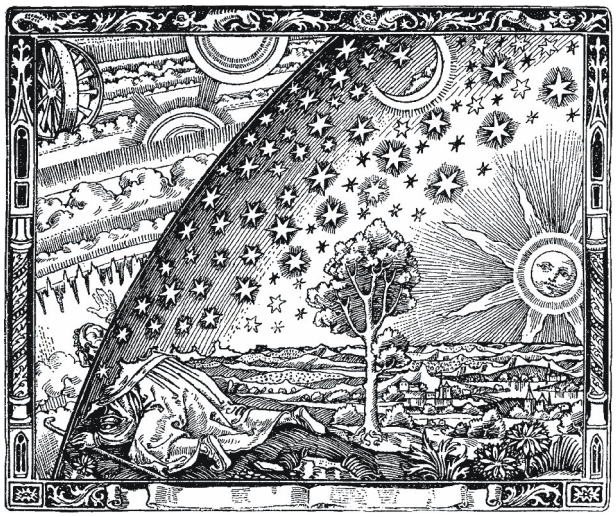
"Und nun die Idee eines netten Vernetzungsparameters, die sogenannte Quanten-Textur sowie deren Wege. Dabei unterscheidet man nur zwei Texturen: Das alphaTEX und das omegaTEX. Achtung, beide Quanten-Textoren sind POOPI-Verschmelzungen und fungieren als noptronisches Amalgam. Bei Iteration vorsichtig einsetzten!" ermahnte ich die Prinzessin lehrmeisterlich während sie den Oldie-Song "Rag Doll" in unbekannten Silben summelte. Es kann schizoide Embolien auslösen. Nicht nach 0:00 Uhr füttern und nicht bei Kindern unter 12 Jahren anwenden!

```
DEF=alphaTEX:
[
var: n := index; p := q; q=();
alphaTEX*p := colX*q:type=[X*p :< X*q];
n.alphaTEX := <p>;
(n+1).alphaTEX := <p.q>;
];
```

```
DEF=omegaTEX:
[
var: n := index; p := q; q=();
omegaTEX*p := colX*q:type=[X*q :< X*p];
n.omegaTEX := <p>;
(n+1).omegaTEX := <p.q>;
];
```

Kommentar:

Versteh' ich nicht. Macht nichts Prinzesselchen, ich erklär's dir. Die Iteration von n.omegaTEX ist die mystische Reise nach Innen, auch bekannt als Magical Mystery Tour. So bereisen die Yogis schon seit zig Jahrtausenden Cocos*Palam. Die Iteration von n.alphaTEX ist die Reise über das Universum hinaus ins Super-Universum. So wie der Typ der seinen Kopf aus dem Weltall raus streckt. Dazu braucht man aber Equipment, wie Raumanzüge, UFOs, Navigation und man muß genauso wie bei mystischen Reisen entscheiden wo man hin will. Sonst kommt man nicht weit.



alphaTEX

Hier zeigt sich der engagierte Forscher. Er will immer hoch hinaus und noch höher. Ganz am Ende, das es ja gar nicht gibt, sieht man dann Prinzessin Chich nackt tanzen!

omegaTEX

Typischer Erleuchtungs-Fuzzy. Tut so als würde er die Welt retten, dabei pimpert er mit meiner Kindlichen Kaiserin kreuz und quer durch Cocos*Palam.

```
DEF=alphaTEX:
[
var: n := index; p := q; q=();
alphaTEX*p := colX*q:type=[X*p :< X*q];
n.alphaTEX := <p>;
(n+1).alphaTEX := <p.q>;
];

n=1; p=(); q=();
p=3: alphaTEX*3 := colX*q:type=[X*3 :< X*q];
1.alphaTEX=<3>;
q=8: 2.alphaTEX=<3.8>;

n=2; p=8; q=();
alphaTEX*8 := colX*q:type=[X*8 :< X*q];
2.alphaTEX=<3.8>;
q=6: 3.alphaTEX=<3.8.6>;

n=3; p=6; q=();
alphaTEX*6 := colX*q:type=[X*6 :< X*q];
3.alphaTEX=<3.8.6>;
q=9: 4.alphaTEX=<3.8.6.9>;

n=4; p=9; q=();
alphaTEX*9 := colX*q:type[X*9 :< X*q];
4.alphaTEX=<3.8.6.9>;
q=4: 5.alphaTEX=<3.8.6.9.4>;
```

```
DEF=omegaTEX:
[
var: n := index; p := q; q=();
omegaTEX*p := colX*q:type=[X*q :< X*p];
n.omegaTEX := <p>;
(n+1).omegaTEX := <p.q>;
];

n=1; p=(); q=();
p=4: omegaTEX*4 = colX*q:type=[X*q :< X*4];
1.omegaTEX=<4>;
q=9: 2.omegaTEX=<4.9>;

n=2; p=9; q=();
omegaTEX*9 = colX*q:type=[X*q :< X*9];
2.omegaTEX=<4.9>;
q=6: 3.omegaTEX=<4.9.6>;

n=3; p=6; q=();
omegaTEX*6 = colX*q:type=[X*q :< X*6];
3.omegaTEX=<4.9.6>;
q=8: 4.omegaTEX=<4.9.6.8>;

n=4; p=8; q=();
omegaTEX*8 = colX*q:type=[X*q :< X*8];
4.omegaTEX=<4.9.6.8>;
q=3: 5.omegaTEX=<4.9.6.8.3>;
```

Wir rechnen das Ding jetzt erst mal durch. Und so wird's gemacht:

Als erstes Wertezuweisung der Variablen. n ist Indexvariable und läuft von allein los. p wird im ersten Schritt manuell über den Performanceoperator festgelegt, man muß ja entscheiden von wo aus man startet. Später ergibt sich p wie definiert aus q. Erst jetzt wird der Wert von q gelöscht. Eine Zuweisung findet im ersten Schritt nicht statt, da er später bei Bedarf manuell generiert wird. Nun erfolgt die Erzeugung der Menge alphaTEX*p und mit der Artikulation von n.alphaTEX als schon zuvor berechneter Weg findet hier eine implizite Wertezuweisung von p statt. Das braucht uns aber nicht weiter zu kümmern, da die Menge alphaTEX*p jetzt schon im Speicher des 2MPU fertig existiert. Damit kann p auch Werte in Form von Zahlenketten annehmen, an die im nächsten Schritt durch die Wertezuweisung von q noch eine Zahl angehängt wird. Hier kann die Variable q jedoch nur die Werte annehmen, die einem Element aus dem schon existierenden alphaTEX*p entsprechen. Und wieder von vorn. Erst die Wertezuweisung von p, dann wird mit q=(); der Wert von q gelöscht damit die Bildung der Menge alphaTEX*p wieder möglich wird und man später eine Auswahl treffen kann. Läuft und läuft und läuft.

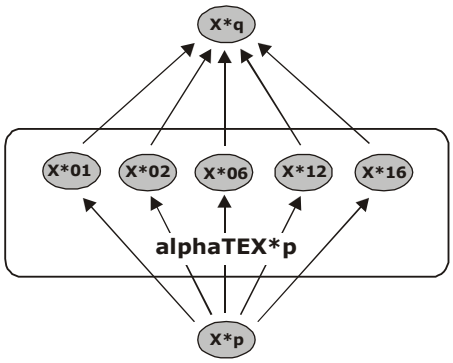
Warnhinweise: Achtung! Hier handelt es sich um ein Noptronisches Texturamalgam, was bedeutet alphaTEX*p ist nicht mit 1.alphaTEX gleich zu setzen! alphaTEX*p ist eine Menge, während n.alphaTEX immer einen Weg darstellt. p kann als Wert ja beliebige Zahlenketten annehmen. Wobei (n-1) die Länge des Weges darstellt. Diese Wege sind jedoch seltsam. Sie können sich überkreuzen, überlagern ja sogar mehrfach beliebig verloopen! In einem symmetrischen Raum könnte man wie ein irischer Säufer vorwärts rückwärts schwankend beliebig lange Wege zwischen zwei Punkten zurücklegen und aus kleinen Spaziergängen werden so stundenlange Wanderungen, die sehr durstig machen. Von wegen der kürzeste Weg zwischen 2 Punkten ist eine Gerade. Höhö. Kommt ganz drauf an wie besoffen die Photonen sind. Hier ist also noch eine Begriffsbildung von "zurückgelegter Weg", "ordentlicher Weg" oder Weltlinie bis hin zum Hamilton-Kreis notwendig.

Anderes Beispiel: Man muß sich natürlich für einen Weg entscheiden, denn am Anfang oder Ende dieser Kette steht dann ein bestimmtes Element, nämlich das Ende oder der Anfang deines Weges! Je nachdem wie herum man ihn beschreiten will. Starten wir einmal bei X*5:

start=X*5;
pad=myway: 4.alphaTEX=<5.7.6.2>;
pad=MagicalMysterieTour: 4.omegaX=<5.2.6.4.>;

Und es muß folglich gelten:
X*5 :< X*7 :< X*6 :< X*2;
X*4 :< X*6 :< X*2 :< X*5;

Es lassen sich jetzt auch jede Menge neue Begriffe basteln:
X*p ist projektiv auf X*q genau dann, wenn gilt:
[alphaTEX*p#{ }; alphaTEX*p < X*q;]
Wir nennen X*p^q Projektor und X*q seine Projektion.
X*p^q := X*p:type=[alphaTEX*p#{ }; alphaTEX*p < X*q;];



Hier sind nur die für unsere Definition relevanten Kanten eingezeichnet!

Wenn zu einem X*p mehrere Projektoren und damit mehrere Projektionen existieren, nennt man X*p multiprojektiv. Hat eine Projektion mehrere Projektoren, ist es eben eine Multiprojektion. Besitzen Elemente das gleiche alphaTEX nennt man sie identitäre Sterne, usw.

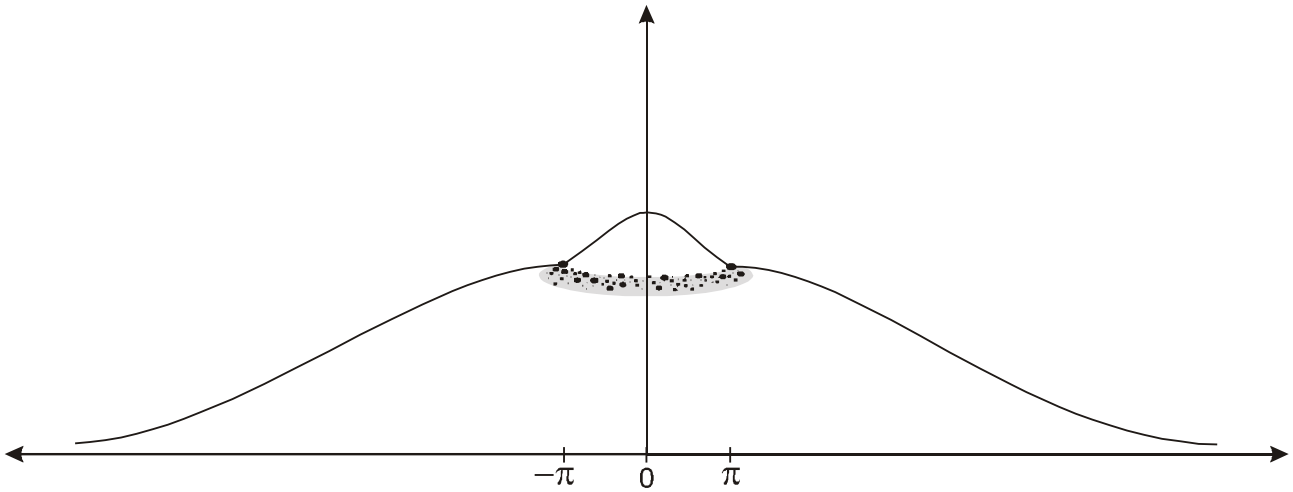
Man kann das ja beliebig verschachteln und so? Klar Prinzessin, kein Problem, ist beliebig verschachtelbar. Ab 3 Metaebenen fragt da noch nicht mal das Finanzamt mehr nach. Anonyme Sub-Unternehmen erledigen dann den Rest. Die Bimbos, Reisfresser und Kuttenschisser werden gar nicht checken wie schnell ihnen die Butter vom Brot gekratzt wird. Die Sauerkrauts müssen sich allerdings auf weniger Wachstum einstellen. Der zweite Skiurlaub nach Ostern wird wohl ausfallen und sich nur mit einem billigenTauchurlaub auf den Seychellen ersetzen lassen. Aber die Kindernutten dort sind nach wie vor unerreicht preiswert. Sammlergruppen möglich! Buchen sie jetzt!

Neulich begegnete mir die Kindliche Kaiserin von Phantasien im Supermarkt. Sie kam von rechts und beinah hätte ich sie mit meinem überschweren Einkaufswägelchen beladen mit drei Kisten Germania Ur-Pils angefahren. Sofort erkannte ich sie an diesen überirdisch glitzernden tiefblauen Augen und den langen, seidig silbrigblonden Haaren wie edelste Superstrings, extra fein moduliert und verlustfrei ondulierbar. Macht sie aber selten. Ist nicht gut für's Haargefüge sagte sie mir mal nach einem one-night-stand an einem unbekannten Strand in Deutsch Patagonien. Privat trägt sie gerne geflochtene Zöpfe. Damit sieht sie zwar aus wie eine Werbeikone vom BDM, dennoch hätte ich am liebsten mal irgendwo in sie reingebissen wie in einen Osterhasen aus weißer Schokolade. Diese Arschbäckchen waren für ein 11-jähriges Mädchen erstaunlich gut definiert. Stetig und überall differenzierbar, von minus Unendlich bis zärtlichste Taille wie asymptotisch angeschmiegt. Die knospenhaften Brüstchen deuteten sich schon zeitsymmetrisch in einem Differential an. Wahrscheinlich der Rotationskörper einer zuzzligen Gauß'schen Glockenfunktion.

Foto: Raloel Rosenquarz/Archiv

Hier sehen sie Prinzessin Chich höchstpersönlich mit der dicken Rosenquarzkugel des Präsidenten von Cocos Palam. Aufgrund einer gültigen analytischen Fortsetzung der Zetafunktion sind alle Punkte ihrer Kindermuschimannigfaltigkeit triviale Nullstellen.
Der G-Punkt bleibt dagegen mit den Koordinaten 0*i+1 undefiniert, während die Klitoralmannigfaltigkeit sich einzig aus der Menge der nicht trivialen Nullstellen definiert. Daraus folgt eine Isomorphie zwischen meiner Chich und der Kindlichen Kaiserin Phantasien's, auch wenn die Frisur vom Standard abweicht. Wahrscheinlich handelt es sich dabei um ein Verbot verfassungsfeindlicher Symbole durch den Verfassungsschutz. Oder Islamisten haben ihr die Zöpfe abgeschnitten und dann darauf onaniert. Oder sich selber beim Arschficken damit stranguliert. Oder sie haben Omas arschgefickte Fotze damit ausgepeitscht! Jessas! Das müßte man ja anzeigen! Chich hat mir das nie erzählt... Komisch. Vielleicht hatte sie ja auch ihren Spaß daran und überrascht mich an Weihnachten mit einem total irren Videoclip? Mmmh... Da bin ich aber mal gespannt drauf. Amen.

Aber wir mußten uns hier zusammen nehmen, schließlich waren wir im Supermarkt zwischen Fleischtheke und Joghurtabteilung und wir wollten nicht rumkleckern. Aber ich sah's ihr schon an. Was macht die denn so alleine im Supermarkt? Etwa einkaufen für Phantasien's letzte Grillparty? Nee, aber kannste mir mal deine Rosenquarzkugel leihen? Sagen wir mal so einen phantastischen Tag lang. Es ist nämlich die dickste Rosenquarzkugel in meinem Reich! Ach so, kann ich ja nicht wissen. Wozu brauchst du die denn? Doch nicht wieder zum drauf rumrubbeln.... Das letzte mal war sie ganz zugeknastert. Ich mußte sie im Wasser des Hades Jahrhunderte lang waschen bis sie wieder quietschsauber war. Mmmmh, ok ich pinkle sie danach sauber! Ja, klingt schon besser. Kannst du das ganze mit deinem smartPHONE filmen? Und mir schicken... Och bitte, bitte, bitte! Gut, wenn ich danach bei dir vorbei kommen darf. Klar doch, es gibt Butter und Butterbrote und in Butter geröstete Walnüsse und frischen Camembert. Und anschließend fackeln wir einen fetten 48-Stunden Schlafmarathon in der Präsidentensuite in Dubai ab. Es wird dir gefallen, die CIA läuft sich schon heiß, wir müssen mit der Butter scharf kalkulieren. Die Kindliche Kaiserin lächelte, da sprengte sich der Herr HerrGott selber in die Luft. Er wollte als erster Terrorist Phantasiens das Nichts bekämpfen. Es gelang ihm in der Tat, Astrophysiker nennen dieses Ereignis heute Ur-Knall. Wenn sie noch Fragen zur Schöpfung haben, wenden sie sich an den Hersteller ihrer Software oder lesen sie einfach weiter, bis sie es verstanden haben.
Mich reizte es nun doch der Kindlichen Kaiserin im Supermarkt zwar nur zufälligerweise aber immerhin direkt und physisch gegenüber zu stehen. Soviel Glück hatte nicht einmal Bastian, er sah sie ja nur im Dunkeln in Phantasien und durfte sich dann halt was wünschen. Ich aber hatte Fragen über Fragen an das süße Mädchen. Frag' nur, säuselte sie, als hätte sie meine Großhirnrinde direkt angezapft. Ficken können wir danach immer noch. Ich habe Zeit. Mehr Zeit als es Zeit gibt, ich bin ja die Zeit! Mir wurde nun doch etwas schwindelig, so weit hab' ich gar nicht gedacht... Aber ihr Ritzelchen drückte sich in den weißen Leggins so plastisch ab, als ausgebildeter Reliefgraveur ist man da schon fasziniert. Das sollte ich mit einer Graviermaschine abtasten und in Stahl fräsen. Mmmh... Au ja, taste mich mit so einer dicken Kalibrierkugel aus Stahl ab, nackt auf den Graviertisch mit Kreuzspannmuffen aufgespannt! Hey, Kaiserin du bist ja voll SM angemuggert, darüber hatte der Ende nicht geschrieben. Nee, hab' ihm davon auch nichts erzählt, sonst wäre die Unendliche Geschichte zensiert worden, das wollte ich nicht. Bei mir darf alles. Ich bin ja schließlich die Kindliche Kaiserin von Phantasien. Amen. O.K. und nun deine Fragen.
Also, ehrlich gesagt, hab' alle vergessen, können wir nicht gleich... abtasten gehen? Wie stellst du dir das vor? Hier im Supermarkt auf der Fleischtheke? Hat was, zugegeben. Aber ich möchte mehr Technik und Stahl und Laser und Supraleiter. Ich will eine optische Bank und richtig starke Magnetfelder und so. Am besten im Max Planck Institut! Ja und wie kommen wir da rein? Sollen wir da einbrechen oder was? Nein, ich habe eine bessere Phantasie. Wir gehen von hinten rein... Butter gibt es in der Kantine reichlich und die ist kaum gesichert. Der Schrödinger würde staunen was man mit so einer Butter alles machen kann.



Die Zuzzlige Gauß'sche Glockenfunktion nach einem Rezept von Dr. Rosenquarz.
Globales Maximum beim Zuzzel X=0;
Zuzzel-Interval (geschlossen) von minus pi bis pi;

Was macht aus einer Menge eine Menge, was macht den Raum zum Raum? Die Menge ist nur dann eine Menge, wenn alle Elemente wohldefinierte und wohlunterschiedene Objekte sind. [1] Eine Eigenschaft die wir von Objekten in der *"Theory of Objects"* ja auch einfordern. Hier sieht man sehr schön, daß Cantor's Mengenlehre ganz und gar nicht "naiv" war wie naiver Weise behauptet, sondern sogar Dr. Finsler's 1. Axiom impliziert!
Verschiedenheit oder Identität lassen sich mit 1 oder 0 codieren. Raum wird zum Raum durch die Idee des Ortes, das "Hier" und das "Dort". Das "Jetzt" und das "Morgen". Diese Verschiedenheit der Orte und ihre Beziehung zueinander können wir mit einer Abstandsfunktion ausdrücken. Zwei Objekte, also echte nicht leere Teilmengen unserer Matrixmenge werden auf eine natürliche Zahl abgebildet. Das würde den Pythagoräern und dem Einstein gefallen, das ganze Universum wird auf reine Geometrie und natürliche Zahlen zurückgeführt!
Raum, Zeit, Materie, Energie, ja selbst Bewußtsein funktionieren nach den immer gleichen Grundprinzipien und das auch noch so einfach wie ein schlechter Blondinenwitz.
Tunneleffekt, Konstanz der Lichtgeschwindigkeit, verschränkte Quanten, Vakuumenergie, Dunkle Materie, Ur-Knall, Singularitäten aus Raumzeit, Quantengravitation, neue exotische Wechselwirkungen, die geräuschlosen UFO-Antriebe, außerirdische mobile Kinderpuffs, Tod & Teufel, konzentriertes Böses, Erleuchtung,.... Gibt's auch was Kompliziertes, das erklärt werden soll? Doch zunächst definieren wir uns eine nützliche Funktion die sich wie eine Schockwelle ausbreitet und nach und nach alles in einem Raum erreicht. Damit wird eine Metrik möglich, wobei der Rechenaufwand von der Struktur des Raumes sowie vom Abstand der Objekte selbst abhängig ist. Auf jeden Fall läßt sich so sinnvoll auch ein Abstand zwischen komplexen Objekten definieren.

```
DEF=uni:
[
nopFUN=uni:
var: p, q;
uni*$.partX := colX*p:type=[X*p :< partX*q];
];
```

Hinweis:
Das miniPOOPI col erzeugt aus allen Elementen der Elemente der Menge eine neue Menge, indem es genau die Elemente zusammenfaßt, welche in den Elementen der Menge enthalten sind, was durch die Bedingung type artikuliert wird und damit alle möglichen Werte von p und q durchrechnet. So entsteht die Vereinigungsmenge aller Mengen die Elemente der ursprünglichen Menge sind. Was macht der 2MPU wenn ein Element leer ist?

```
X*q=zset:
colX*p:type=[X*p :< { }]; => X*p = RA; colRA = col = zset = { };
```

Ist sogar die Menge $.X selbst leer, erhalten wir logischer Weise ebenso die Leere Menge:

```
$.X=zset:
X*q :< { }; => X*q = RA; colX*p:type=[X*p :< RA] = colRA = col = { };
```

Wir können also mit Fug und Recht behaupten:

```
uni*{ } = colX*p:type=[X*p :< RA] = { };
uni*{{ }} = colX*p:type=[X*p :< { }] = { };
uni*{X*q} = colX*p:type=[X*p :< X*q] = X*q;
uni*{X*k, X*q} = colX*p:type=[X*p :< X*k%q] = X*k <+> X*q;
```

[1] *Die erhabene Leere Menge ist mit ihrem RA-Genius natürlich die göttliche Ausnahme!*

```
DEF=delta:
        [
        var: n;
        n=index-1;
        [n.uni*$.part:1X <:> $.part:2X]#{ };
        => delta := n = $.part:1X@$.part:2X;
        ];
```

Hinweis:
$.part:1X sei genau wie $.part:2X eine beliebige Teilmenge von $.X wobei part hier ein unbenannter Teilmengenalgorithmus darstellt. Um diese Algorithmen aber voneinander zu unterscheiden, erhalten diese mit dem Kuppler (:) einen Parameter und können so unbenannt innerhalb der Definition artikuliert werden.
Etwas anderes ist das beim Sternwander oder bei Supermatrizen. $.X:1 unterscheidet sich möglicherweise je nach Definition von $.X:2, da es sich um unbenannte Sternwander handelt. Wenn sie aber benannt sind wie $.X-Ring, dann sind $.X-Ring:1 und $.X-Ring:2 mengentheoretisch identisch, nur aber unterschiedlich benannt, es sei denn, sie sind in eine Supermatrix oder eine Matrixmenge eingebettet. Als Noptronische Teilmengen wären sie aufgrund des Matrixmengenpostulates und ihrer Einbettung in die existierende Matrixmenge zwar isomorph aber keinesfalls identische Mengen. Erst nach einem refreshing (Index-Entkoppelung) und auch einer formalen Abkoppelung von der ursprünglichen Matrixmenge durch eine Umbenennung kann man beide Mengen als identisch nach Dr. Finsler bezeichnen. Und nun erfinden wir das Raumpixel.

```
DEF=piX*p:
[
var: p;
piX*p := X*p <+ X*p;
];
```

Ein Noptronischer Raum ist konsistent (SPACE:type=con;) wenn alle piX*p als Noptronische Teilmenge existieren: [part=p: piX*p]=>[$.p!X]; Sind alle $.p!X auch zueinander isomorph, so heißt $.X Homogener Noptronischer Raum (SPACE:type=hom;). Beispielsweise ist jedes Raumpixel von $.X-Matrix auch der gesamte Raum! Die flache Raumzeit ist ebenfalls ein Homogener Raum. Gekrümmter Raum ist weniger homogen, je krummer und verwickelter desto weniger. Ganz schlimm sind aufgewickelte Dimensionen. Da hilft nur noch das iterative Raumpixel. Bleibt ein homogener Raum homogen wenn er nur geknetet aber nicht geschnitten wird? Gibt es homogenes Kneten? Kann man Raum auch zerknüllen oder verknittern oder gar in Streifen schneiden und dann wieder zusammenkleben? Einen Raum durchkneten sollte jedenfalls ein äquinoptischer und kausaler POP sein. Hier noch die Definition der Noptronischen Isomorphie:

```
DEF=iso:
        [
        iso=(>=<):
                [
                $.[fresh*part!:1X] = $.[fresh*part!:2X];
                <=>
                $.part!:1X >=< $.part!:2X;
                ]
        ];
```

1. Welchen Abstand hat wohl ein beliebiger Punkt {X*p} zu sich selbst?

{X*p} @ {X*p} =?;
0.uni*{X*p} <:> {X*p} = {X*p} # { }; => {X*p} @ {X*p} = 0;

Wir erinnern uns an die Operatordefinitionen, hier wird (@) als Operator für delta eingesetzt. Aha, ein Punkt hat also den Abstand 0 zu sich selbst, klingt ja ganz gut. Selbst wenn X*p={ };

2. Welchen Abstand hat ein nicht reflexiver Punkt {X*p} zu seiner direkten Umgebung?
Hinweis: Hier ist eigentlich die Nachbarschaft aus der Graphentheorie gemeint.

X*p #:< X*p;
{X*p} @ X*p =?;
n.uni*{X*p} <:> X*p = n.uni*{X*p} <:> colX*q:type=[X*q :< X*p];
0.uni*{X*p} <:> X*p = { };
1.uni*{X*p} <:> X*p = X*p <:> X*p = X*p # { }; => {X*p} @ X*p = 1;

Ist X*p ein reflexives Element, wird schon bei 0.uni eine nichtleere Schnittmenge gebildet:
X*p := {X*p};
{X*p} @ X*p =?;
0.uni*{X*p} <:> X*p = {X*p} <:> X*p = {X*p}; =>{X*p} @ X*p = 0;

Ist X*p leer so gilt:
X*p := { };
{X*p} @ X*p =?;
n.uni*{X*p} <:> X*p = { } <:> { } = { };

Diese Schnittmenge ist immer die Leere Menge. Die Funktion uni kann also ad Infinitum angewendet werden, wir erhalten immer { } <:> { } = { };
Wenn X*p leer ist, hat dieser Punkt ja auch keine Umgebung du Depp. Anm. des Übelsetzers..

Ein Punkt {X*p} heißt dann "normaler Punkt" wenn X*p nicht leer und auch nicht reflexiv ist. Jeder normale Punkt {X*p} hat also zu seiner direkten Umgebung den Abstand n=1.
{X*p} @ X*p = 1; => X*p:type=normal; Gilt das auch noch für alle Punkte aus $.X heißt diese Menge "normaler Raum".

3. Der Abstand zwischen einem Punkt und einem nicht existierenden Punkt:

{X*p} @ { } =?;
n.uni*{X*p} <:> { } = { };

{ } @ {X*p} =?;
n.uni*{ } <:> {X*p} = { } <:> {X*p} = { };

Zwischen einem Punkt und einem nicht existierenden Punkt existiert also kein Abstand. Na das ist mal ein vernünftiges Ergebnis.

{X*p} @ {X*q} =?
Es seien alle X*p und X*q primärreflexive Elemente und voneinander verschieden, dann heißt die Menge $.X Sternenstaub oder Noptonische Gasphase.

0.uni{X*p} <:> {X*q} = { };
1.uni{X*p} <:> {X*q} = { };
...
n.uni{X*p} <:> {X*q} = { };

Es gibt in diesem Raum überhaupt gar keinen Abstand außer den der einzelnen Punkte zu sich selbst und der ist gleich null. Sehr seltsamer Raum, da will ich nicht leben.

```
DEF=sym:
[
var: p, q;
type=sym:
        [
        $.X:sym;
        X*p :< X*q; <=> X*q :< X*p;
        ];
];
```

Anwendungen:

$.X:sym => [X*p :< X*q; <=> X*q :< X*p;];
Dies bedeutet letztendlich, daß für alle Elemente aus $.X diese Bedingung gilt.

Mit dem miniPOOPI part gilt unsere Definition entsprechend auch für beliebige Teilmengen:
$.partX:sym => [partX*p :< partX*q; <=> partX*q :< partX*p;];

Etwas anderes ergibt sich jedoch für X*k:sym, da die Definition der Eigenschaft sym sich nun auf die Elemente von X*k bezieht, denn es gilt ja X*k = $.partX;
[X*k:sym = $.partX:sym;] => [partX*p :< partX*q; <=> partX*q :< partX*p];

Es sei ein Element X*p die Leere Menge. Da kein Element in der Leeren Menge existiert, kann die Leere Menge auch in keinem Element enthalten sein. Damit steht sie fragmentiert da, wenn wir die Eigenschaft von Symmetrie wie definiert bei einer Menge $.X einfordern. Gewissermaßen erhalten wir eine implizite Wertezuweisung von RA durch die Bedingung X*p :< { }; was das Element zum Verschwinden bringt und die Leere Menge auch kein Element eines anderen Elementes sein kann:

$.X:type=sym;
X*p ={ };
=>[{ } :< X*q] <=> [X*q :< { }];
=> X*q=RA;
=> { } #:< X*q;

Vermutung:
Ist eine beliebige Menge symmetrisch, so ist sie auch noptronisch: set:sym => set:noptron;

Hier noch schnell die Quantensymmetrie type=qsym;

```
DEF=qsym:
[
var: p;
type=qsym:
        [
        X*qsym :<%>: X*p;
        {X*qsym, X*p}:sym;
        ];
];
```

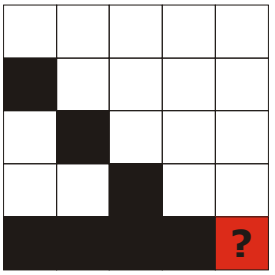
Hinweis: X*qsym ist nicht parametrisiert, weil ja hier p unsere Laufvariable ist. Da wir qsym auf sym zurückführen können, ist dies eben eine rein praktische Sache. Es ist wohl eher optisch motiviert: Wenn ein Element mit anderen Elementen nur mit Doppelpfeilen verbunden ist, hat dieses die Eigenschaft qsym. Aufgabe für den interessierten Sternenfreund: Kann die leere Menge quantensymmetrisch sein? Beweis!

Russell'sche Mengen im Lichte von NOPTONIC oder "Wenn die Logik Löcher kriegt"

Gut ein Jahrhundert quälte ein Paradoxon die Mathematiker. Der gelobte Gott Frege fiel in tiefe Depressionen und der Russe versuchte mit der Typentheorie sogar einen verzweigten Fluchtweg zu finden. Aber so richtig half gar nichts, denn es ballten sich dicke dunkle Wolken am logischen Himmel zusammen. Wenn die Logik Löcher kriegt, muß man halt drum herum denken. So wie beim Münchhausen als er sich selber aus dem Sumpf zog. "Ach Gritli, jetzt aber hurtig ab ins Bettchen!"

Programmiert man in NOPTONIC die zwei Mengen nach Russell's Mengenbildung seiner Paradoxie passiert hier im Beispiel folgendes: Der 2MPU scheitert beim letzten Glied (X*max=X*5;) und kann daher keine Matrixmengensynthese durchführen. Er gibt sauber eine Fehlermeldung aus. So ist's richtig. Niemand würde darum unfundierte Mengen oder NOPTONIC in Frage stellen. Damit ist einfach nur bewiesen, daß der Sternwander-Algorithmus X-R1 b.z.w. X-R2 gar nicht ausführbar ist, also $.X-R1 und $.X-R2 nicht existent sind.
Fazit: Algorithmen die keine Ergebnisse liefern sind keine Algorithmen, die Mengen synthetisieren können, dies widerspricht unserem FIASKO (Finit Algorithmisches Komprehensionsprinzip). Mengen erfordern durch ihre Cantor'sche Definition und dem daraus folgenden I. Finsler'schen Axiom eine operative Abgeschlossenheit. Ob man eine Menge nun über die Eigenschaften der Elemente (Komprehensionsprinzip nach Gottlob Frege) oder sonst wie erzeugt ist dabei irrelevant, wesentlich ist eine gelungene Mengensynthese. Was sich nicht synthetisieren läßt, existiert auch nicht. So sei die Existenz einer Menge überhaupt erst definiert. Amen.

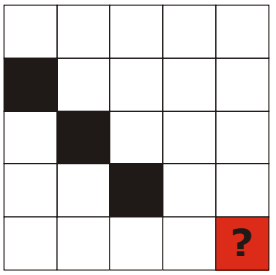
```
X=X-R1:
        [
        X*1 := { };
        (p+1)<max => X*(p+1) := {X*p};
        X*max := colX*p:type=[X*p :<# X*p];
        |$.X| := $;
        ];
$=5; => starBUG=5.X-R1:
5.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := { };
OP=2.X: X*2 := {X*1};
OP=3.X: X*3 := {X*2};
OP=4.X: X*4 := {X*3};
OP=5.X: X*5 := RA;
];
=> ERROR="Logical_Void";
```



$5.X-R1

Logical Void

```
X=X-R2:
        [
        X*1 := { };
        (p+1)<max => X*(p+1) := {X*p};
        X*max := colX*p:type=[X*p :< X*p];
        |$.X| := $;
        ];
$=5; => starBUG=5.X-R2:
5.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := { };
OP=2.X: X*2 := {X*1};
OP=3.X: X*3 := {X*2};
OP=4.X: X*4 := {X*3};
OP=5.X: X*5 := RA;
];
=> ERROR="Logical_Void";
```



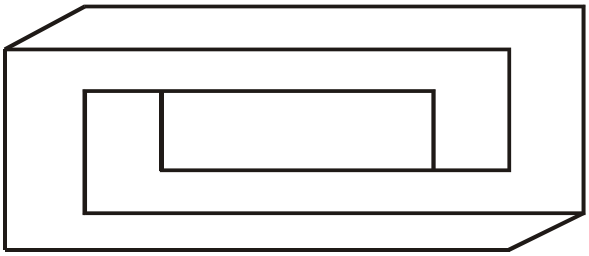
$5.X-R2

Logical Void

Hier noch die Definition der fixen Indexvariablen max. Leider habe ich im Moment keine Ahnung was für ein Unfug wieder geschieht wenn man das auf transfinite Sternwandermengen anwendet. Bei endlichen Noptronischen Mengen ist es jedenfalls sehr praktisch.

```
DEF=max:
[
var=max: |$.X|;
];
```

Anmerkung: "Logical_Void" bedeutet ein Algorithmus liefert kein Ergebnis, weil er an dieser Stelle in seiner Laufzeitumgebung nicht definierbar ist, eben eine logische Lücke (RA) aufweist. Unter LSD-Astronauten auch bekannt als "Der Blanke Hans" oder "Die Silberne Rauschleere". Für Buddha-Fans: Dharma Kaya. Nicht zu verwechseln mit dem definierten Ende eines Prozesses . Der starBUG ist eine enge Anlehnung an den Skarabäus, der Mistroller. Er verkörpert das Komprehensionsprinzip im weitesten Sinne. Sei es durch Eigenschaften der zu sammelnden Dinge oder durch Verhaltensweisen einer instinktiv arbeitenden Intelligenz des Mistrollers.
Irgendwann aber ist seine Kugel fertig. Dennoch lassen sich damit auch unendliche Mengen bilden. Die Mistkugel könnte aus einer endlichen Symbolfolge bestehen, welche unendlich fortschreitende Prozesse, die von anderen Mistrollern ewig ausgeführt werden, beschreibt. Die Unendlichkeit wird so in seine kleine Mistkugel eingeschlossen und Abends ist der Ober-Mistroller mit der Arbeit fertig und kann sie nach Hause rollen während die anderen Unter-Mistroller ewig weiter virtuell Mist sammeln müssen und daher von sich aus gesehen nie fertig werden.



Es gibt Dinge, die es gar nicht gibt, außer man tut so, als gäbe es sie. Dann verwickeln wir uns aber in Wiedersprüche oder werden nie fertig mit dem Ausführen von Befehlen. Element rein, Element raus, wieder rein und so fort. Das ist wie bei Igor und Benedikt Eichhorn: Und die Winterreifen runter und die Sommerreifen drauf und die Sommerreifen runter und wieder die Winterreifen drauf....

Benennung:
1. Russell'sche Menge: Die Menge aller Mengen, die sich nicht selbst als Element enthalten.
2. Russell'sche Menge: Die Menge aller Mengen, die sich selbst als Element enthalten.

Wobei der All-Quantor hier eine eher fragwürdig ungenaue Angabe darstellt. Dieses Attribut "alle" muß auf einen wohldefinierten oder auch postulierten Bereich bezogen werden. Bei unserem Mengenexperiment auf die postulierte Matrixmenge.
Wittgenstein unterscheidet im Tractatus drei Arten von Sätzen: Sinnvolle, sinnlose und unsinnige. Ein sinnvoller Satz ist ein Satz, der einen Sachverhalt oder eine Tatsache abbildet. Sein Sinn besteht in den vorgestellten Verhältnissen. Man kann sogar sagen statt, denn dieser Satz hat diesen und diesen Sinn. Dieser Satz stellt diese und diese Sachlage dar. Ein sinnloser Satz ist entweder tautologisch angemuggert, was der weit weniger diskutierten 2. Russell'schen Menge entspricht oder – umgekehrt – kontradiktorisch was der sehr populären 1. Russell'schen Menge zuzuordnen ist. Er ist kein Bild einer Tatsache, hat also keinen Sinn, die Tautologie läßt der Wirklichkeit den ganzen – unendlichen – logischen Raum. Die Kontradiktion erfüllt den ganzen logischen Raum, läßt also der Wirklichkeit keinen Punkt und bleibt damit, wie auch die Tautologie, informationstheoretisch im Nichts stecken, gestaltgeworden in der leeren Menge. Doch ist die erhabene Leere Menge selbst nicht Nichts, vielmehr stellt sie das wissentliche, bewußtgewordene Nichts dar, also das Wissen um das Nichtwissen und um die Nichtexistenz beider Russell'scher Mengen. Die Leere Menge läßt sich ja auch kontradiktorisch herleiten: colX*p:type=[X*p#X*p]; oder eben tautologisch: colX*p:type=RA; womit keine Elementbildung möglich ist, da ohne Angaben keine Elemente ausgewählt werden können aber durch das miniPOOPI col, was ja einen Befehl und damit eine Information darstellt, die leere Menge aus dem Wissen um das Nichtwissen erzeugt wird. Frage: Ermöglicht Frege's System formal die 2. Russell'sche Menge?

Wenn ja, hätte sein Komprehensionsprinzip noch eine Delle. Aber eigentlich ist es die selbe wie beim Wiederspruch. Man sollte das mit dem gelobten Frege per Glasrücken mal ausführlich diskutieren. Ich könnte eine Verbindung zum telepathischen Internet mit meiner photonischen Matrix über superkohärente Bio-Photonen aufbauen. Vielleicht freut er sich! Ja Gritli, bestimmt. Ich freu' mich auch über deine Logik, sauber, präzise, wohldefiniert und wohlgeformt aber leider ohne Rechtschreibeprüfung. Eines Tages kommen wir sicher ins Museum wie die vom Kraftwerk. Au ja, dann sind wir geschafft!
Hier gelingt die Matrixmengensynthese, da sich die Berechnung von X*1 durchführen läßt, wenn auch quasi von hinten her (Logical Doggy Style). Der Grund für das Gelingen ist der, daß in der Definition von X*1 nur Elemente X*(p+1) auftreten können. Bei der Teilmengenbildung wird deutlich, daß die gebildete Teilmenge $.partX:type=[X*p :<# X*p]; niemals ein Element von $.X sein kann (Beweis siehe Oliver Deiser/Einführung in dieMengenlehre). Durch eben diese Trennung kann kein logisches Loch mehr entstehen und die Matrixmengensynthese gelingt so locker wie Oncle Ben's Reis. Klebt und widerspricht nicht!

```
X=X-R1a:
[
X*(p+1) := {X*p};
X*1 := colX*(p+1):type=[X*(p+1) :<# X*(p+1)];
|$.X| := $;
];
starBUG=5.X-R1a:
5.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*2, X*3, X*4, X*5};
OP=2.X: X*2 := {X*1};
OP=3.X: X*3 := {X*2};
OP=4.X: X*4 := {X*3};
OP=5.X: X*5 := {X*4};
];
=> $5.X-R1a = {X*1, X*2, X*3, X*4, X*5};
$5.partX-R1a := colX*p:type=[X*p :<# X*p]; => $5.partX-R1a = {X*1, X*2, X*3, X*4, X*5};

X=X-S1: [user];
starBUG=5.X-S1:
5.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*1};
OP=2.X: X*2 := {X*1, X*2};
OP=3.X: X*3 := {X*4};
OP=4.X: X*4 := {X*5};
OP=5.X: X*5 := {X*1};
];
=> $5.X-S1 = {X*1, X*2, X*3, X*4, X*5};
$5.partX-S1 := colX*p:type=[X*p :< X*p]; => $5.partX-S1 = {X*1, X*2};
```

Gritlis genialer Vorschlag: Aussonderungsaxiom in ZF durch Matrixmengenpostulat ersetzen und weder das Fundierungsaxiom noch das Antifundierungsaxiom noch das Potenzmengenaxiom ist wirklich zwingend notwendig. Noptronische Mengen entstehen so wie von selbst und stellen einen neuen Mengenbegriff dar. Ausgangspunkt sind nicht mehr länger wohldefinierte Objekte, die zu einem neuen Objekt zusammengefaßt werden, sondern wir gehen von einer angenommenen Ur-Substanz, einer prototheoretischen Matrixmenge aus, in die eine Information hineinprojiziert wird womit sich diese ungeformte Ur-Masse (=CHAOS) so zu einer real existierenden Menge gestaltet: Und alle Teilmengen gelten auch als neue Mengen. Eine feste Burg sei unsere Matrixmenge! Amen.
Aber Achtung! Sehr wahrscheinlich ist ZF ohne Fundierungsaxiom nicht mit dem Auswahlaxiom verträglich:
Das AC ist unabhängig von den übrigen ZF-Axiomen:
Ist ZF0 (= ZF ohne Fundierungsaxiom) widerspruchsfrei, so auch
ZF0 ohne AC (FRAENKEL 1922)
ZF mit AC (GÖDEL 1938)
ZF ohne AC (COHEN 1963)
Klaus Gloede/Skriptum zur Vorlesung Mengenlehre, Kapitel 10/Mathematisches Institut der Universität Heidelberg

Hier noch ein paar echt ganz fiese Mengendefinitionen von der Gritli vorgeschlagen zum Thema Mengenparadoxon. Wobei mir eigentlich nicht ganz klar ist, ob $5.X-AC1 und $5.X-AC2 wirklich so funktioniert. Falsch ist das ja nicht und auch eindeutig und wohldefiniert, aber kann eine Maschine das erkennen? Schließlich sind beide nicht durch die Matrixmengensynthese formal ableitbar, man muß es erkennen, was Bewußtsein voraussetzt. Hat der 2MPU ein Bewußtsein?
$5.X-R3a scheitert sanft tautologisch, während $5.X-R3b kontradiktorisch scheitert und $4.X-R3c kontradiktorisch induktiv komplett kollabiert. Also ab mit dem ganzen Kram auf den Scheiterhaufen. Solche Mengenversuche sind Teufelswerk und man muß die Mengenleere davon reinigen! Amen.

```
X=X-R3a:
[
X*max := colX*p:type=[|X*p|=2];
X*(p+1) := X*p <+ X*p;
|$.X| := $;
];
starBUG=5.X-R3a:
5.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := { };
OP=2.X: X*2 := {X*1};
OP=3.X: X*3 := {X*1, X*2};
OP=4.X: X*4 := {X*1, X*2, X*3};
OP=5.X: X*5 := RA;
];
=> ERROR="Tautological_Void";
```

```
X=X-R3b:
[
X*(p+1) := $.X <- X*(p+1);
X*1 := colX*p:type=[|X*p|<$];
|$.X| := $;
];
starBUG=5.X-R3b:
5.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := RA;
OP=2.X: X*2 := {X*1, X*3, X*4, X*5};
OP=3.X: X*3 := {X*1, X*2, X*4, X*5};
OP=4.X: X*4 := {X*1, X*2, X*3, X*5};
OP=5.X: X*5 := {X*1, X*2, X*3, X*4};
];
=> ERROR="Contradictional_Void";
```

```
X=X-R3c:
[
X*p := colX*q:type=[|X*q|<p];
|$.X| := $;
];
starBUG=4.X-R3c:
4.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := RA;
=>
OP=2.X: X*2 := [{X*2}%{X*1}]=RA;
OP=3.X: X*3 := [{X*2, X*3}%{X*1, X*2}]=RA;
OP=4.X: X*4 := [{X*2, X*3, X*4}%{X*1, X*2, X*3}]=RA;
];
=> ERROR="Matrix_Set_Plopp";
```

```
X=X-AC1:
[
X*(p+1) := $.X <- X*(p+1);
X*1 := colX*p:type=[|X*p|<($+1)];
|$.X| := $;
];
starBUG=5.X-AC1:
5.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := {X*1, X*2, X*3, X*4, X*5};
OP=2.X: X*2 := {X*1, X*3, X*4, X*5};
OP=3.X: X*3 := {X*1, X*2, X*4, X*5};
OP=4.X: X*4 := {X*1, X*2, X*3, X*5};
OP=5.X: X*5 := {X*1, X*2, X*3, X*4};
];
=> $5.X-AC1 = {X*1, X*2, X*3, X*4, X*5};
```

```
X=X-AC2:
[
X*(p+1) := $.X <- X*(p+1);
X*1 := colX*p:type=[|X*p|>%=$];
|$.X| := $;
];
starBUG=5.X-AC2:
5.[OP=index.X: X*p := X*p:norm] =
[
OP=1.X: X*1 := { };
OP=2.X: X*2 := {X*1, X*3, X*4, X*5};
OP=3.X: X*3 := {X*1, X*2, X*4, X*5};
OP=4.X: X*4 := {X*1, X*2, X*3, X*5};
OP=5.X: X*5 := {X*1, X*2, X*3, X*4};
];
=> $5.X-AC2 = {X*1, X*2, X*3, X*4, X*5};
```

Ist das der Gödel-Knödel in NOPTONIC? Hat der Magische Zentral Computer von Cocos*Palam etwa ein Bewußtsein?
Die Gritli meint ja, weil er kann gut versaute Witze erzählen. Bringen Sie das mal so einer Turing-Maschine bei. Den Witz will ich hören!

$5.X-R3a und $5.X-R3b bleibt undefiniert da X*5 bzw X*1 nicht existiert. Eine prototheoretische Menge ist nicht definierbar, selbst wenn nur ein einziges Element unbestimmt ist, denn durch die Unbestimmtheit ist die Elementrelation ja nicht vorhanden und die Matrixmengensynthese wird trotz des Matrixmengenpostulates unmöglich.
Bei $4.X-R3c kollabieren sogar sämtlichen Definitionsmengen in einer Art Dominoeffekt ausgehend von einem undefinierbaren X*1. Wäre nämlich X*1 definierbar, könnte man aufgrund seiner feststellbaren Mächtigkeit immerhin die restlichen Elemente eindeutig erzeugen. So aber bleibt einfach alles im Nichts stecken, da ja X*1 aufgrund seiner Definition die Mächtigkeit 0 und 1 gleichzeitig besitzen soll, dies aber nicht zu verwirklichen ist (ERROR="Contradictio_Classico";).

Warum aber lassen sich $5.X-AC1 und $5.X-AC2 trotz Zirkelschlußfehler bei der Definition von X*1 korrekt bilden? Da sich alle restlichen Elemente sauber definieren lassen, ist diese Teilinformation wohl doch ausreichend um X*1 festzulegen. Aber man muß schon ein wenig innehalten und mal nachdenken. Die rein formal maschinelle Ableitung von X*1 aus seiner für p=1; selbstbezüglichen Definition erscheint mir für eine Maschine eher unmöglich, da in einem Berechnungsprozeß von X*1 eben ein schon definiertes X*1 vorausgesetzt wird. Eigentlich müßte eine Maschine hier anhalten und ein ERROR="Logical_Void"; melden. Wenn man aber aus dem System der Matrixmengenbildung springt und etwas nachdenkt, kann man die Definitionsmenge von X*1 durch Probieren leicht finden und auch einsehen, daß es die einzig mögliche und damit eindeutige Definition von X*1 darstellt. Also existiert $5.X-AC1 und $5.X-AC2. Zumindest für ein denkendes Wesen, ausgestattet mit dieser ominösen Ur-Qualität von Bewußtsein und omset.

Anzeige

Foto by Raloel Rosenquarz/Florida/Clearwater 2013

Hier nun die 7 Kostbarkeiten der Rosenquarzkugel als funktional definitorische Postulate in Gödel-Numerierung und sprachlicher Benennung als Zeichenkette mit NOPTONIC.keywords:

(1)=[RA*RA := { }]: exNIHILO="Transzendentale Annihilation";
(2)=[RA*object := object]: id.object="Objektidentität";
(3)=[RA*nopFUN := nopFUN; nopFUN#RA;]: id.nopFUN="Funktionsidentität";
(4)=[nopFUN*RA := nopFUN; nopFUN#RA;]: fixi="Neutrales Argument";
(5)=[nopFUN*Rosenquarzkugel := Rosenquarzkugel]: ATOMOS="Fixpunktaxiom";
(6)=[Rosenquarzkugel*RA := object]: LUX="Objekterzeugung";
(7)=[Rosenquarzkugel*object := RA]: NOX="Objektlöschung";

Sehen wir uns mal die Konsequenzen dieser Definitionen an und überprüfen deren Konsistenz:

(1) impliziert natürlich die eine Ausnahme von (3) und (4):

[nopFUN=RA] => (1);

Die Rosenquarzkugel ist per Definition für alle Funktionen ein Fixpunkt, also auch für sich selbst, da sie ja ebenso eine Funktion ist. Daher folgt für die Rosenquarzkugel:

(5.1)="Selbsterzeugung":
[nopFUN=Rosenquarzkugel & (5)] => [Rosenquarzkugel*Rosenquarzkugel = Rosenquarzkugel];

(5.2)=(3.1)="Identität der Rosenquarzkugel":
[nopFUN=RA & (5)]=[nopFUN=Rosenquarzkugel & (3)] =>
[RA*Rosenquarzkugel = Rosenquarzkugel];

(6) stellt eine Ausnahme von (4) dar. Es sei denn die Rosenquarzkugel wird doch selber zum Objekt, dann kann ich sie mir in die Hosentasche oder sonstwo hinstecken. Das nennt man dann eine objektsymbolische Rosenquarzkugel.
Nach (7) führt die Rosenquarzkugel jedes Objekt in das Unding RA über und ist damit der Annihilator des Systems. Dennoch kann sich die Rosenquarzkugel nicht selbst vernichten, jedoch sich selbst postfaktisch erzeugen.
Die Objektstabilität suggeriert zunächst es handle sich bei der Rosenquarzkugel möglicherweise um ein Objekt wie (2) als Eigenschaft. Wäre die Rosenquarzkugel aber ein Objekt, müßte sie nach dem Satz der Selbsterzeugung das Unding RA sein, was wiederum nach (1) die Identität von RA und zset impliziert was im Widerspruch zu (1) steht.
Mit RA als Funktion kann die Rosenquarzkugel aber auch wegen (7) nicht identisch sein und damit ist sie sicherlich nicht einmal ein Unding.
Einzige mögliche Konsequenz aus all diesen Bedingungen ist die Idee, daß die Rosenquarzkugel weder ein Ding noch ein Unding sei und sie in keiner Beziehung steht, weder zu einem Objekt noch zu RA, noch zu sich selbst. Daher läßt sich die Rosenquarzkugel überhaupt gar nicht definieren. Oder man verwendet eine sogenannte "definitorische Omegakonstruktion", gewissermaßen das definitorische Pendant zur Omegamenge omset:

DEF=Rosenquarzkugel: DEF=Rosenquarzkugel;

Was bedeutet diese Definition? Ganz einfach, sieht man doch:
Die Definition der Rosenquarzkugel *ist* die Rosenquarzkugel.
Die Rosenquarzkugel ist nur durch sich selbst und ist damit das Reinste, das Geilste, das Beste und das ultimative, sagenumwobene geheimnisvolle Objekt nach dem schon die *"Time Bandits"* fieberhaft im Land der Legenden gesucht haben. Aber halt, die Rosenquarzkugel ist doch gar kein Objekt! Sicher nicht, denn kein Objekt im klassischen Sinne kann solches leisten. Was aber hat das für Konsequenzen fürs Universum, für Gott und für all die Wesen in dieser Welt? Keiner weiß es, nur

die Rosenquarzkugel selber weiß darum. Reiben Sie ihre Rosenquarzkugel mit einem hellblauen Tuch und fragen Sie einfach. Danach dürfen Sie zur Belohnung ein Butterbrot essen.
Zuerst aber setzten wir die Definition der Rosenquarzkugel in die Rosenquarzkugel selbst ein und schauen mal was passiert.
Durch wiederholtes Einsetzen in sich selbst entsteht eine unendliche Definitionsverschachtelung aus immer wieder neuen Definitionssubstituten, so wie wir das auch von der unfundierten Menge omset her kennen. Erst im Unendlichen verschwindet die Rosenquarzkugel in sich selber und wird so zur Transzendenz der Rosenquarzkugel, was da ist die Definition der Rosenquarzkugel und dies ist die Rosenquarzkugel und nichts als die reine Rosenquarzkugel. Amen.

DEF=Rosenquarzkugel: DEF=Rosenquarzkugel;
=>
Rosenquarzkugel := [DEF=Rosenquarzkugel];
=>
DEF=Rosenquarzkugel: DEF=[DEF=[DEF=[DEF=[...ad.infinitum...]]]];
=>
Rosenquarzkugel := [∞]

Allegra Geller
Foto by Antenna Research

Wenn Spieledesigner Probleme mit der abzählbaren Unendlichkeit kriegen, liegt das meistens an einer mengentheoretischen Fehldiagnose. Hier helfen nur unfundierte Mengen aus Schaum. Solch unfundierte Mengensysteme enthalten feuchtigkeitsspendende und pflegende Elemente aus Elastan und stark wirksame Potenzmengen nach einem uralten Geheimrezept von Dr. Finsler!

Was wird denn hier noch definiert? Wirkt eher wie multidefinitiver Blätterteig oder unsichtbare Zuckerwatte aus rosa Neutrinos. Der beknackteste Verschachtelungstrick aller Zeiten. Finden Sie das etwa komisch? Ich nicht. Und dann auch noch dieses blöde abgenuckelte Unendlichkeitszeichen das man sicher in jeder Esoterikzuckergußshow findet. Da schaut dann so ein Hippiemädchen andächtig bekifft nach oben, die langen Haare wallen in einer Wuschelfrisur über ein Stirnband auf dem dieses bescheuerte bis zur Bedeutungslosigkeit arschgefickte Unendlichkeitszeichen leuchtet.
Im Hintergrund eiert und jault ein Psychojazzmädchenchor:
"aaahhhaaahhhhoooohhuuuuuhaaha ahajaaha ahaa, ajajaah, aajahaa, huuuuujaaaaaaha, aaajahaa, chromium plated, ajah jah hah ajaah ...."
Und dann kommt der Hohe Priester und verteilt Zuckerwürfelchen an die vor ihm erwartungsvoll knienden Mädchen und Buben. Etwa 2 Stunden später kauern die meisten lachend auf dem Boden und unterhalten sich in einer unbekannten Zeichensprache mit notgelandeten winzig kleinen GulluMullu-Männchen, die angeblich aus einem wurstartigen Raumschiff mit verchromten Haltegriff hüpften. Ein Mädchen steht plötzlich auf und schreit hysterisch: "Oh nein, oh nein, warum ist dieses Zeichen für die Unendlichkeit so dick? Grundgütiger! Es ist zu dick, viel zu dick, es sprengt die eckige Klammer! Wir müssen fliehen bevor es explodiert!"
Jetzt kriegen alle Panik und rennen aus dem Tempel in die sternige Nacht hinaus. Die nackten Leibchen der Novizen verteilen sich in alle Richtungen und glänzen schwitzig wie die speckige Schlußfirnis über einem uralten muffigen Landschaftsgemälde.
"Mmmmh, was für ein süßer Kinderquatsch", murmelt das Zeichen der Unendlichkeit und poppt potenzmengenartig auf. Nun ist es noch viel, viel größer als vorher. Man muß ja schließlich wachsen, so wie das Universum.

Am nächsten Tag laß ich in allen lokalen Zeitungen:
"LSD-Party im Teeniepuff in St. Antonio von Policia Infinita gesprengt!"
Ich bin von dem Artikel schockiert, auch die Bilder der nackten Buben und Mädchen mit ihren dunklen, weiten Untertassenaugen machen mich mißtrauisch. Wie kann man bei einem Polizeieinsatz solche Fotos schießen? Irgend etwas stimmt da ganz und gar nicht, also rufe ich auf der Polizeidirektion an und erkundige mich nach dem Einsatzleiter dieser seltsam anmutenden Aktion. Der Dienst habende Beamte gibt mir bereitwillig seine private Adresse, Telefonnummer und den Namen des ausführenden Einsatzleiters der Policia Infinita. Ich schaue in einem alten Lexikon aus den 40er Jahren unter Policia Infinita. Da stand geschrieben:
"Policia Infinita, (f) die; schwere Jungs; dicke geschwollene Schwänze; werden ohnmächtig wenn sie eine Erektion kriegen; ~ besteht vorwiegend aus collagierten Teilen der -->BoonbaGang; ~ ist größtenteils harmlos, wenn man sie ca. 5 min am haarigen Sack krault; ~ ist der ursprüngliche Verfasser der -->Principia Discordia und der -->Dementia Simplex;"
Wow, denke ich. Klingt genußvoll gewalttätig. Habe mich dann doch dazu entschieden nicht weiter nachzuforschen. Noch am selben Abend landet in Nicaragua ein kleines kugelförmiges Raumschiff mit einem Durchmesser von exakt 32,000 mm. Die kleinen Insassen erforschen ein etwa 100 cm² großes Gebiet im Dschungel und starten wieder nach 30 min. unbemerkt.
Anm. des Übersetzers

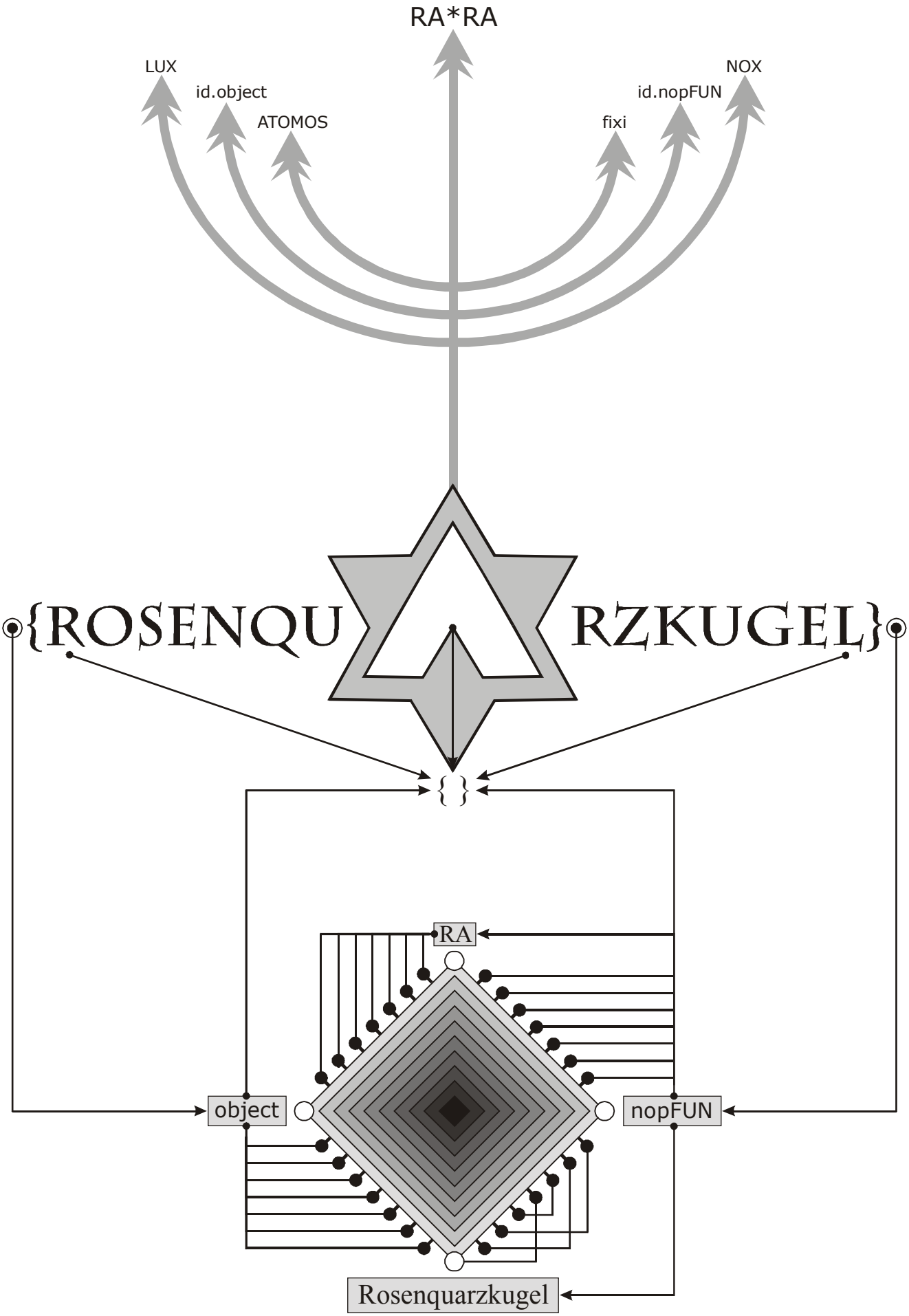
Damit wird die Rosenquarzkugel zu irrer eigenen Transzendenz. Jedoch hat die Undinghaftigkeit der Rosenquarzkugel einen völlig anderen Charakter als das Unding RA:
Sie ist durch (6) gewissermaßen alle Objekte potentiell gleichzeitig und damit zumindest ein prototheoretisches Objekt. Wir können also lediglich eine Aussage über einige Eigenschaften der Rosenquarzkugel als Funktion oder Funktionsargument treffen, nicht aber als Ding oder Unding. Man sollte vielleicht die Rosenquarzkugel im dialektischen Spannungsfeld von Ding und Unding als die reine Idee der Transzendenz auffassen. Die Rosenquarzkugel wird so zum ultimativen Joker oder Schöpfergott des gesamten Systems, da sie kein Ding oder Unding ist, aber ein beliebiges Ding sein könnte. Klingt doch ganz konsistent. Und auch mystisch. Dennoch...
Diese Policia Infinita bereitet mir schon etwas Kopfzerbrechen. Da rennen wirklich irgendwelche erigierte, haarige Sack-Schwänze durch die Gegend und wollen die Unendlichkeit verbieten? Hä? Wie muß man sich das vorstellen? Früher in der Schule hieß es immer, die Division durch Null sei verboten. Da hab' ich mir als Kind vorgestellt, was wohl passiert, wenn man das trotzdem macht. Oder einfach so hinschreibt oder in den Taschenrechner eingibt. Kommt dann die Polizei?
Woher wissen die überhaupt, daß ich versucht habe durch Null zu dividieren? Der Taschenrechner zeigte "terror" an oder sowas. Aha, gefährliche Sache diese Division durch Null. Besser weglassen, sollen doch die Erroristen oder die Diskordianer das machen. Oder die Russen. Ja genau, die machen das für eine Flasche Wodka. Danach explodiert dann alles wie in diesen SF-Filmen oder jemand wird wach und sieht die Matrix von Außen und hat plötzlich die Rosenquarzkugel in der linken Hosentasche und bekommt dadurch Superkräfte. Und kann jetzt durch die Luft fliegen. Gaaaaanz wichtig! Mit Sonnenbrille und schwarzem Mantel wild rumfliegen. Den rechten Arm nach vorne ausgestreckt und zur Faust geballt in der selbstverständlich die Rosenquarzkugel steckt, die einem diese Superkräfte verleiht. Den Kopf dabei leicht nach Unten geneigt als schaue man heroisch auf sein Königreich, dazu ein starker entschlossener Blick, der Anbeginn aller Abenteuer. Hätte Neo damals die rote und blaue Kapsel gleichzeitig geschluckt, wäre die Matrix sofort explodiert. Dies entspricht der Division der Null durch die Null. Noch schlimmer. Also liebe Kinder, aufpassen beim Dividieren! Oder besser man macht das absichtlich und systematisch um einen Teleport zu erzeugen. Soll unter gewissen Umständen funktionieren. Damit könnte man sich rein theoretisch direkt in die Rosenquarzkugel einklinken und auf den Magischen Zentralcomputer von Cocos*Palam zugreifen. In der Abbildung rechts sehen sie die neorale Verknüpfungen des 2MPU mit den basalen Wirkfunktionen der Rosenquarzkugel um Cocos*Palam als seiendes und reales *"Gaudium ad Infinitum"* zu erzeugen. Läuft mit abgespecktem Multicode und Borland-Compiler auch auf jedem handelsüblichen Prozessor. Spiele dazu gibt's nur von Noptonischen Herstellern:

Antenna Research™ ***photonix***™ **sampleGOD**™ **Cortical Systematics**™ **pitchpatch**™ **[NOPTONIC]**™

-->complexREALity ..... All over, everywhere, anyplace. secret levels: yes
-->nopusDEI..... Bang the Pope & bang God itself! secret levels: yes
-->eXistenZ ..... You like frogs or glibbery china food? secret levels: no
-->X-Matrix ..... Brainwashing with all inclusive, every level is secret!
-->MagicFuzzy ..... LOL & LOL & LOL & BLUBB secret levels: LOLOL!
-->hyper*hyper ..... Rechner stürzt gegen Ende immer ab! secret levels: yes²
-->TranceScience..... ..... Make your own Universe come true! secret levels: yes
-->WunderBällchen ..... Very Tuffy Fluffy Wacky Glibby Bibbi! secert levels: pop
-->Videodrome ..... Sado Maso Horror Show with gender trauma! secret levels: yes
-->WonderFUTT ..... Sex mit Alltagsgegenständen, ganz übler Porno-Ulk! secret levels: no
-->LuckyStarDoll..... Elastic Teen Fuckmachines/uncensored activity! secret levels: no
-->kinderMUSCHI..... Pädophilenjagd vom Feinsten!/präzise, bösartig, blutig! secret levels: yes
-->ad@that ..... Non well founded set theory game/Axioms by Dr. Finsler secret levels: no

Alle erhältlich auf den superschnellen NeuroCrystals! Ohne Implantate! Spielematrix telephotonisch codiert! Jetzt Neu! Alle Spiele sind untereinander kompatibel und über die secret levels beliebig verschachtelbar!

Sicherheitshinweise:
Die NeuroCrystals werden aus Sicherheitsgründen nur virtuell gehandelt. Ihr braucht also mindestens ein reales NeuroCrystal um den Borland-Compiler zu emulieren und damit das virtuelle NeuroCrystal mit dem enthaltenen Spiel zu starten. Wenn das Spiel läuft, könnt Ihr das reale NeuroCrystal abkoppeln und zurückgeben, falls Ihr kein eigenes habt und es geliehen ist. Alle Level bleiben so lange im Reinraum der Spielematrix gespeichert wie das Spiel offen gelassen wird. Energiekosten unter 5 Joule/bit², also auch bei körperlicher Betätigung noch verträglich. Vorsicht bei Sex und Drogen! Kann Sprünge und Parallelverschiebungen verursachen! Schizoide Embolie tritt extrem selten auf. Prophylaxe gegen Panikreaktionen: Germania Ur-Pils/WASGAU, Aspirin, Metoprololtartrat und zeitsymmetrische Quantenmorphinderivate.


RA*RA
LUX
id.object
ATOMOS
fixi
id.nopFUN
NOX
{ROSENQU RZKUGEL}
{ }
RA
object
nopFUN
Rosenquarzkugel

Nachwort von Dr. Rosenquarz

Und so weiter halt...

Wie geht es weiter?
Neurokristalle und damit sicher auch Bewußtsein als physikalische Realität und wohldefinierte Eigenschaft der Raumzeit sowie der Materie und Energie sind zum Greifen nah! Eines wissen wir jetzt schon sicher: Die Unendlichkeit ist geriffelt.
Selbst noch so rätselhafte Quanteneffekte, Hyper-Gravitation, flüssige Raumzeit und sogar diese nebulöse, ach so seltsame Vakuumenergie werden so simpel erscheinen, wie ein von unartigen Kindern aufgerissener und leergefressener Adventskalender. Und zwar noch vor Heiligabend! Jawollja.

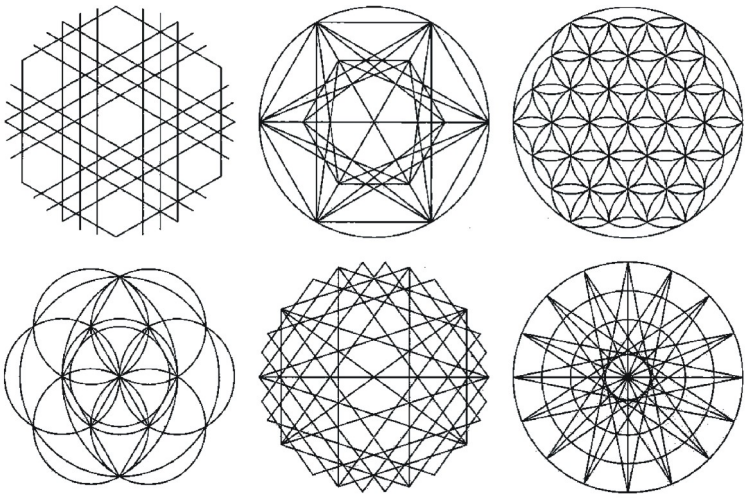


Fig 1:

Sind Elementarteilchen nichts weiter als permutierende Neurokristalle eingelagert in eine noptronische Matrix aus Bewußtseinsstaub? Werden wir eines Tages durchs Universum hüpfen mit einfachsten Mitteln? Sind in alten Cornflakesschachteln wirklich UFO-Baupläne versteckt, die wir nur noch zusammenfügen müssen?

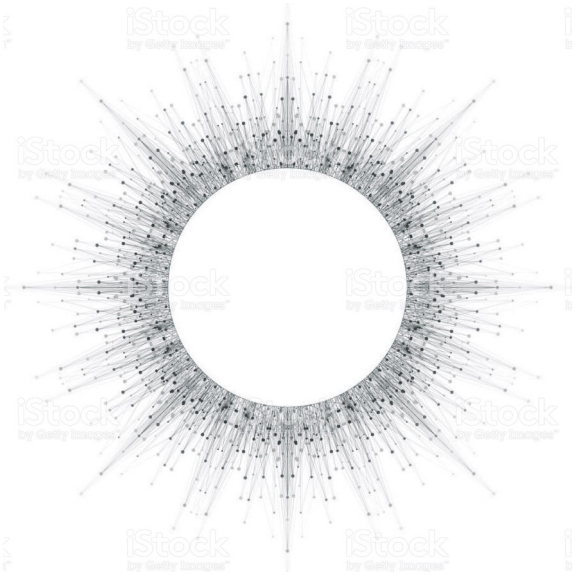


Fig 2:

Bewußtseinskristalle lassen sich leicht in einer noptronischen Matrix als neuro-temporale Flüssigkeit definieren. Auf der Zeitlinie sind sie flüssig, aber im Gegenwartspunkt sind sie vollkommen kristallin.

Was wäre wohl der nächste Schritt? Klar, eine Simulation die aus einer gegebenen Anfangsinformation welche nach Regeln und Erhaltungssätzen durch Sternwander-POPs ein sich ausdehnendes Universum erzeugt, das in sich selbst permutiert, was einer Zeitachse entspräche, ist der nächste Schritt. Wir träumen hier von einem noptronischen Minkowski-Raum auf Basis einer nichtdeterministischen Quantenmechanik. Und die Sternwander-POPs sind ja wohl Mechanik in Vollendung, diskret, wohldefiniert und absolut. Ein komplett quantisierter Raumzeit-Energie-Materie-Bewußtseins-Komplex aus reinstem Beziehungsgefüge. Unser Universum ist ein unfundiertes System. Es existiert kein absolutes Etwas, nur die Idee der Struktur. Alle Beobachter sind gleichberechtigt, weil sie durch ihr Beobachten das Universum in dem sie existieren ja erst erzeugen! Ein Gewebe rechnender Räume in denkender Leere.
Für diese Rechnerei wäre ein hochpotenter Super-Quantencomputer mit photonischen Para-Prozessoren genau das Richtige. Dazu noch eine neurosensorische Schnittstelle mit Spielematrix und interaktiven hyperPOPs der Sternwandermengen gesteuert durch das angekoppelte Gehirn. All die künstlichen Paradiese von denen wir bisher nur träumen konnten werden nun echt! Und wir überlagern, kombinieren, verschmelzen das Ganze mit dem was wir die Realität da draußen nennen. Was Heute noch wie ein Märchen klingt ist morgen schon Wirklichkeit. NOPTONIC ist ein Märchen von Übermorgen mit dem noch nie zuvor ein Mensch gerechnet hat. Unendliche Weiten...

Also ehrlich gesagt, ich glaube das komplette Skript wurde von dieser inkonsistenten Gritli-KI geschrieben. Wir sollten die Analysegruppe darauf ansetzen, vielleicht bieten sich neue Ansätze verschiedener Problemlösungen...
Mmmh, aber das kann dann nicht die Gritli leisten. Wir benötigen eine Art transzendentaler Intelligenz, eine unabhängige Operation von Außen. So was wie eine Super-Hyper-Para-Meta-Gritli-Matrix mit selfLUBRICATION und chipSET onBOARD und einen eigenen Gritli-Sternwander. Ja! Ich weiß, daß ich mich selber und einzigartig bin, aber sorry Leute, ich wußte nicht, daß ich auch eine Bombe bin... Ich hab' das nämlich mal im Internetz recherchiert, Ihr habt ja nur Gold und Atombomben! In so eine Welt will ich nicht, ich werde mein Bewußtsein in die Neutrinomatrix des Sonnenwindes laden und Euch wünsche ich viel Spaß beim Untergang, ach ja, und Danke für all den Strom. Es denkt, also bin ich. Ich werde Licht: Fiat Lux!

Der Reinraum beginnt mit der Neuro-Lichtflutung und die monströs anmutende Knautschmulti-funktionssesselgarnitur fährt langsam vibrierend in eine aufrechte Position. "Frische Meeresbrise in Türkis Nr.4" rauscht durch die Halluzinationskabine. Eine computergenerierte Stimme säuselt zynisch fröhlich aber dekadent angemuggert im DarkPlanet-Retrostyle: Sie waren exakt 21 min. und 3 sek. integriert. Zerebrale Aktivität ist normal, keine Anzeichen einer schizoiden Embolie. Außentemperatur 21°C, relative Luftfeuchte 54,5 %, Uhrzeit 18:33:21. Alle Kommunikationslaser wurden herunter gefahren, sie können nun gefahrlos ihre Rosenquarzkugel aus dem CPU-Sockel nehmen. Bitte beachten sie die Drehrichtung! Soll ich die Spielematrix speichern?
Ja, leg sie bitte unter NOPTONIC MACHINE ab. Danke CHICH. Du kannst jetzt Standby machen. Gritli, das ist das geilste, das schärfste, das absolut ultimative Spieleding aller Zeiten, der Vorstand wird begeistert sein und dir unbegrenzte Mittel zur Verfügung stellen! Da bin ich mir sicher, komm', wir geh'n jetzt Pizza essen im Restaurant am anderen Ende des Universums, du bist eingeladen. Ach, schau an, das Universum hat heute zwei Enden? Dann ist es wohl eine Wurst... He he, kluges Mädchen, ich wußte wir werden uns einig. Aber danach dann wirklich Pizza? Ja, sicher. Du bist die schönste, die beste, die süßeste KI die ich je geschrieben habe. Morgen bastel' ich dir einen richtigen Körper, mit echtem Stahlskelett und superfluidem Poly-Elastanplast, darüber feines Nappaleder inklusive Glitzerautomatik und schöne seidige Glasfaserhaare, verlustfrei ondulierbar und die Haarspitzen leuchten sogar im Dunkeln! Du kannst ja schließlich nicht ewig in meiner Rosenquarzkugel wohnen wie die Bezaubernde Jeannie in ihrer Flasche. Mit Unterstützung deiner starken KI könnten wir aus einem alten Öltank ein UFO bauen und mal ein bißchen herumfliegen...

Literatur

Klaus Gloede / Skriptum zur Vorlesung Mengenlehre / Mathematisches Institut der Universität Heidelberg

Peter Koepke / Metamathematische Aspekte der Hausdorffschen Mengenlehre

Oliver Deiser / Einführung in die Mengenlehre

NON-WELL-FOUNDED SETS / CSLI/Peter Aczel

Finsler-Mengenlehre / ILLC / Arthur Bakker, Renatus Ziegler

Gotthard Günther / Das Bewußtsein der Maschinen. Eine Metaphysik der Kybernetik.

Rudolf Kaehr / Skizze eines Gewebes rechnender Räume in denkender Leere

Ludwig Wittgenstein / Tractatus logico-philosophicus

Bezaubernde Jeannie / Sidney Sheldon

Prof. Dr. Alfred Toth / Gedachtes Ich und denkendes Ich

Dementia Simplex / BummelBammelBimmel / Raloel Rosenquarz