Aristotle, Boole, and Chu
Duality since 350 BC

Vaughan Pratt
Stanford University

August 27, 2015

George Boole Mathematical Sciences Conference
University College Cork

DUELITY

PART 1

ARISTOTLE

Aristotle's syllogisms

Nobody is despised who can manage a crocodile.

Illogical persons are despised.

Therefore illogical persons cannot manage crocodiles.

($\frac{2}{3}$ of Lewis Carroll's first of 60 sorites)

Subject S: *illogical*

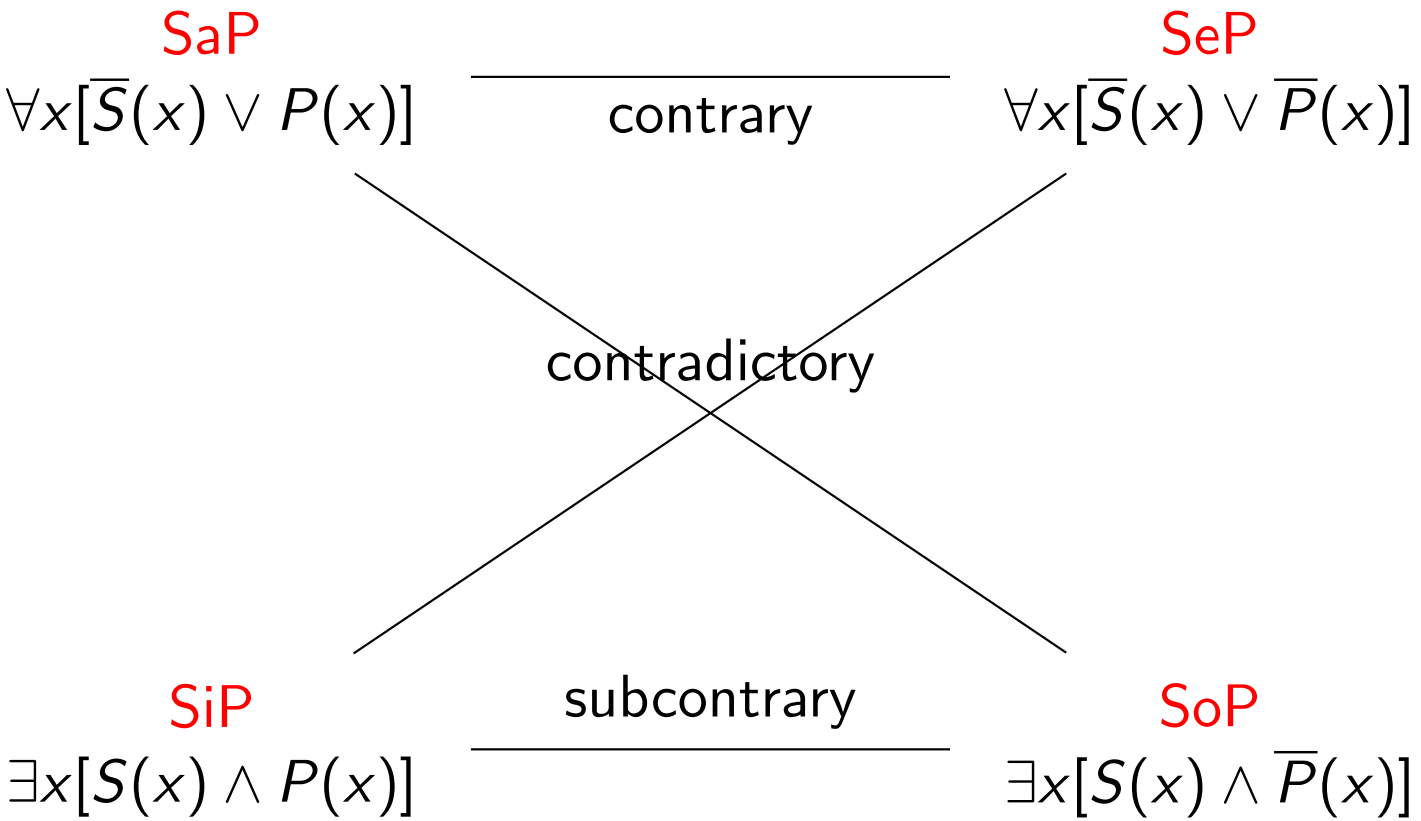
Predicate P: *can manage a crocodile*

Middle term M: *is despised*

Form: PeM, SaM $\vdash$ SeP. (EAE-2)

Major premise J, minor premise $N \vdash$ conclusion C.

Square of Opposition



Duality of top and bottom rows: $\forall\vee$ dual to $\exists\wedge$.

Syntax

A syllogism is a sequent $J, N \vdash C$ consisting of major and minor premises J and N and a conclusion C.

Major premise J: MaP, MeP, MiP, or MoP (4), or converse (+4) (fig+1).

Minor premise N: SaM, SeM, SiM, or SoM (4), or converse (+4) (fig+2).

Conclusion C: SaP, SeP, SiP, or SoP (4), converse disallowed (+0).

Mood: The three connectives, e.g. MeP,SiM $\vdash$ SoP has mood EIO.

Figure: 1 plus any increment from converting either premise.

Form: mood-figure, e.g. MeP,MiS $\vdash$ SoP has form EIO-3 (minor converse adds 2).

Hence $8 \times 8 \times 4 = 256$ possible forms.

Semantics

Syllogistic sentence: $\forall x[L_1(x) \vee L_2(x)]$, L_i's distinct literals
. . . or its negation, i.e. $\exists x[L_1(x) \wedge L_2(x)]$.

Syllogistic set: set of syllogistic sentences every pair of which has at most one literal in common. (Literals compared only up to sign, so no LEM.)

Theorem

Any syllogistic set S with at most one universal sentence is consistent.

Proof.

Let u be the universal sentence of S if any.
Model of S: universe $E =$ the set of existential sentences.
For each $e \in E$, for each literal L in e, or in u but not e, set $L(e) = T$.
Set the remaining values of literals to T. □

This construction fails for syllogistic sets with ≥ 2 universal sentences.
Exception: all sentences universal ($E = \emptyset$), hence vacuously consistent.

Corollary

A valid syllogism must contain exactly one particular among its 2 premises and contradicted conclusion. (Hence only $\frac{3}{2^3} \times 256 = 96$ possible forms.)

Proof.

Two particulars $\rightarrow$ one universal, refuted by Theorem 1.
If no particulars the empty universe is a counterexample. $\square$

Theorem

A syllogism $J, N \vdash C$ is valid if and only if its translation $\hat{J} \wedge \hat{N} \wedge \neg \hat{C}$ into **propositional** *calculus is unsatisfiable.*

Proof.

By the corollary any counterexample requires only one individual.
But then $\forall x$ and $\exists x$ act as the identity operation, so can be dropped.
Each $L_i(x)$ then simplifies to L_i, giving propositional calculus. $\square$

So to decide validity of a form, just treat it as propositional calculus.

Program to enumerate the valid forms

```
#include <stdio.h>
#define S 0xaa
#define M 0xcc
#define P 0xf0
#define T 0xff
unsigned char pred[3][2] = {{M, P}, {S, M}, {S, P}};
char rel[2][2] = {{'A', 'E'}, {'I', 'O'}};
main()
{
  int sp, i, j, m, n, c;
  unsigned char J, N, nC;
  for (sp = 0; sp < 3 ; sp++) {
    for (i = 0; i < 2; i++)
      for (j = 0; j <= 0xff; j += 0xff) {
        J  =  sp==1? pred[0][i]  & (j^pred[0][1^i]):
                  (T^pred[0][i]) | (j^pred[0][1^i]);
        for (m = 0; m < 2; m++)
          for (n = 0; n <= 0xff ; n += 0xff) {
            N  =  sp==2? pred[1][m]  & (n^pred[1][1^m]):
                       (T^pred[1][m]) | (n^pred[1][1^m]);
            for (c = 0; c <= 0xff; c += 0xff) {
              nC  =  sp==0?  pred[2][0]   & (T^c^pred[2][1]):
                   (T^pred[2][0])  | (T^c^pred[2][1]);
              if ((J & N & nC) == 0)
        printf("%c%c%c-%d\n", rel[sp==1][j&1],
                                  rel[sp==2][n&1],
                                  rel[sp!=0][c&1],
                                  2*m + i + 1);
}}}}}
```

Commented source at `http://boole.stanford.edu/syllenum.c`

The 15 valid forms, as enumerated by the program

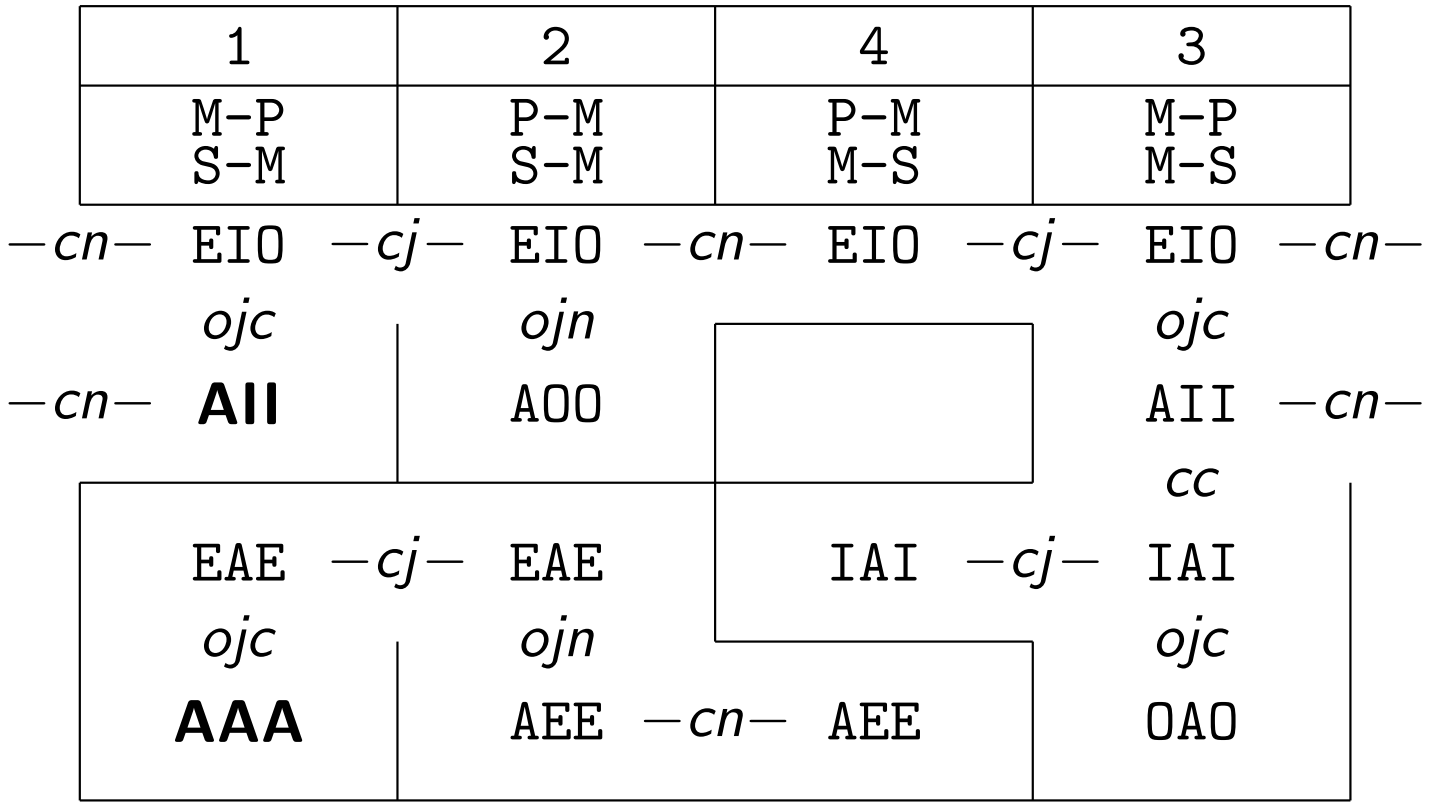
```
AAA-1
EAE-1
AEE-2
AEE-4
EAE-2
IAI-3
OAO-3
IAI-4
AII-1
AII-3
EIO-1
EIO-3
AOO-2
EIO-2
EIO-4
```

The 15 valid forms, organized

1	2	4	3
M–P S–M	P–M S–M	P–M M–S	M–P M–S
EIO	EIO	EIO	EIO
AII	AOO		AII
EAE	EAE	IAI	IAI
AAA	AEE	AEE	OAO

Axioms A1. **AAA-1**
A2. **AII-1**

The 15 valid forms, organized and derived



Axioms A1. **AAA-1**

A2. **AII-1**

Rules R1. Convert one *e* or *i* form: *cj*, *cn*, *cc*.

R2. Obvert two sentences with the same RHS: *ojn*, *ojc*.

The problem of existential import

All ungulates are mammals.

All unicorns are ungulates.

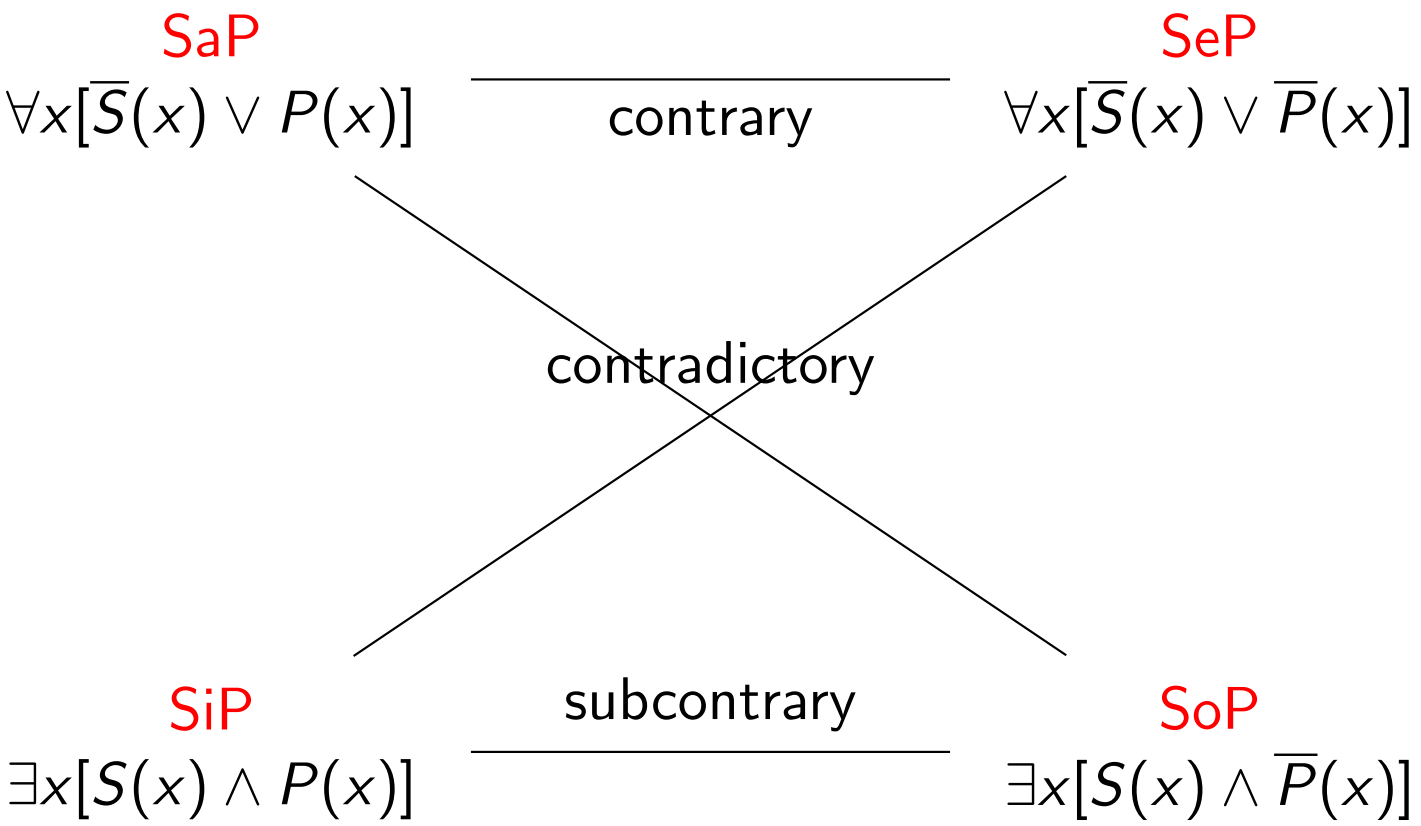
Therefore some unicorns are mammals.

This has form AAI-1.

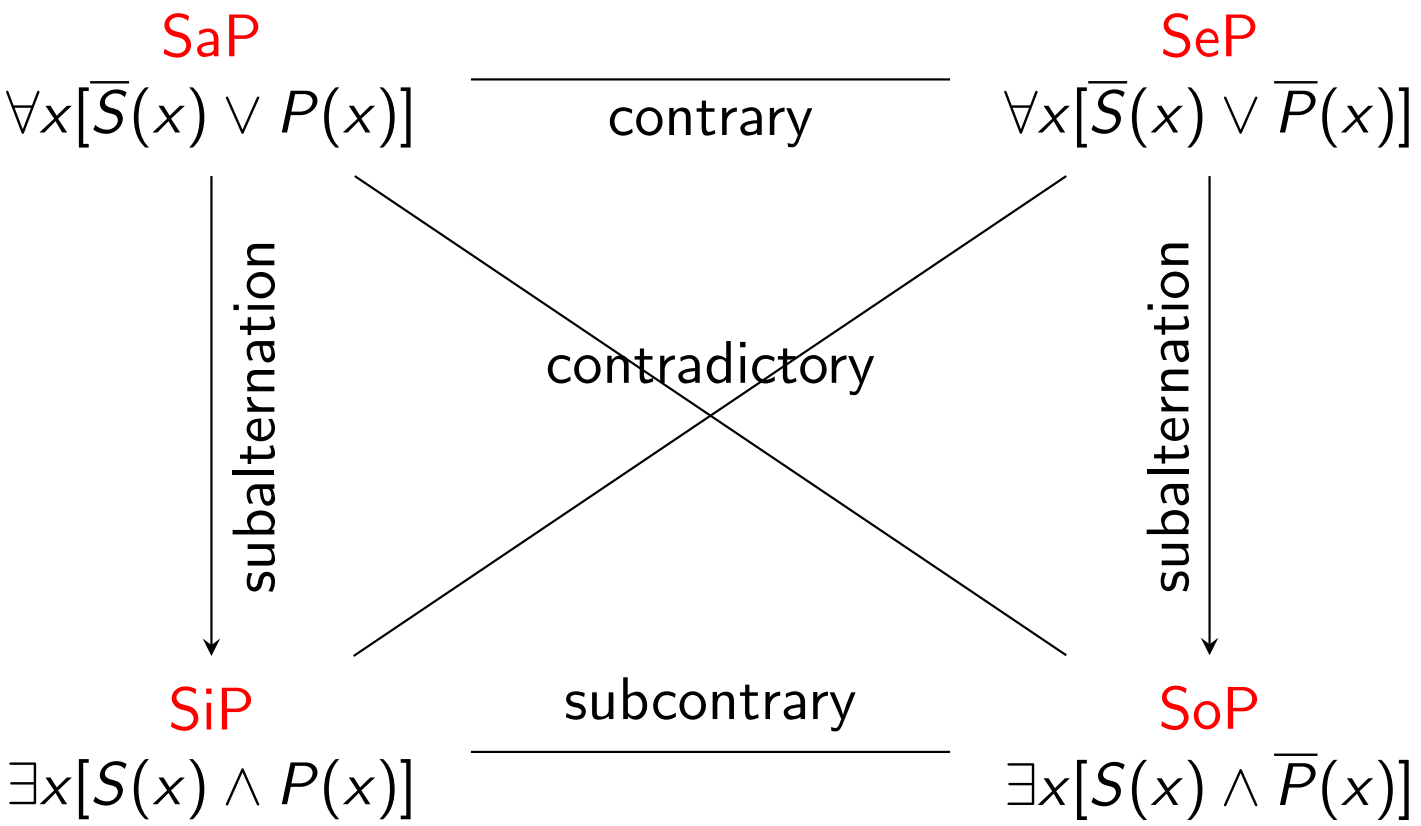
Aristotle considered this form valid, justified by declaring the minor premise "all unicorns are ungulates" to be false in the event unicorns did not exist.

Based on this convention, Aristotle introduced subalternation: when both premises are universal, the conclusion of a valid syllogism may be weakened to a particular.

Square of Opposition



Traditional Square of Opposition



Conditionally valid syllogisms

Nine additional syllogisms are judged valid, conditional on nonemptiness of a suitable one of their three terms.

Hence 24 (un)conditionally valid syllogisms, six in each figure.

Example: Ungulate a Mammal, Unicorn a Ungulate $\vdash$ Unicorn i Mammal.

MaP, SaM $\vdash$ SiP. AAI-1

AAI-1 is *conditionally* valid. Condition: the class S (unicorns) is nonempty.

Using the same rules as before, three additional axioms (for a total of five, A1-A5) suffice to axiomatize these 24 syllogisms.

Conditionally valid syllogisms

Nine additional syllogisms are judged valid, conditional on nonemptiness of a suitable one of their three terms.

Hence 24 (un)conditionally valid syllogisms, six in each figure.

Example: Ungulate a Mammal, Unicorn a Ungulate $\vdash$ Unicorn i Mammal.

MaP, SaM $\vdash$ SiP. AAI-1

AAI-1 is *conditionally* valid. Condition: the class S (unicorns) is nonempty.

Using the same rules as before, three additional axioms (for a total of five, A1-A5) suffice to axiomatize these 24 syllogisms.

Question: Can rules based on subalternation be formulated to reduce the number of axioms?

Conditionally valid syllogisms

Nine additional syllogisms are judged valid, conditional on nonemptiness of a suitable one of their three terms.

Hence 24 (un)conditionally valid syllogisms, six in each figure.

Example: Ungulate a Mammal, Unicorn a Ungulate $\vdash$ Unicorn i Mammal.

MaP, SaM $\vdash$ SiP. AAI-1

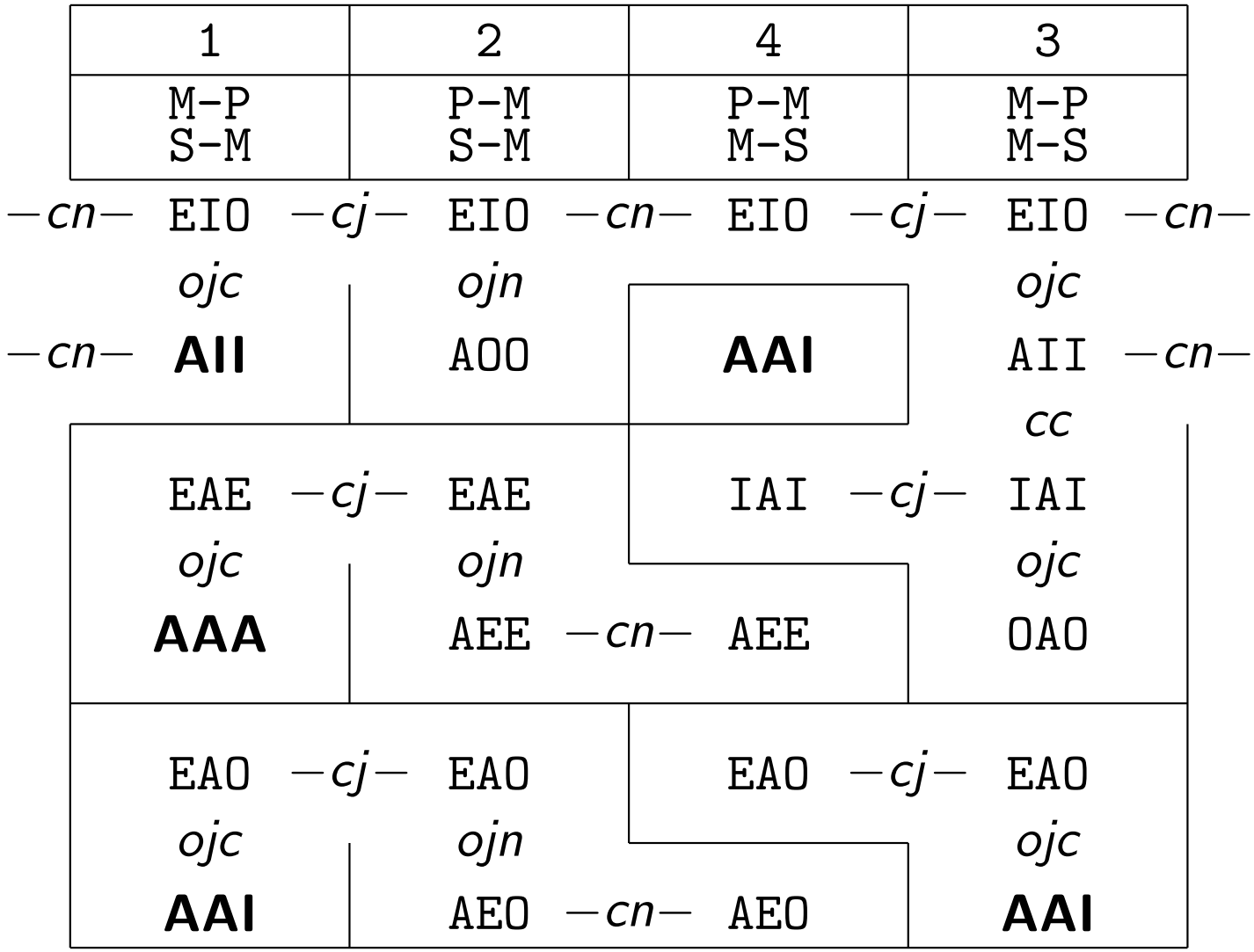
AAI-1 is *conditionally* valid. Condition: the class S (unicorns) is nonempty.

Using the same rules as before, three additional axioms (for a total of five, A1-A5) suffice to axiomatize these 24 syllogisms.

Question: Can rules based on subalternation be formulated to reduce the number of axioms?

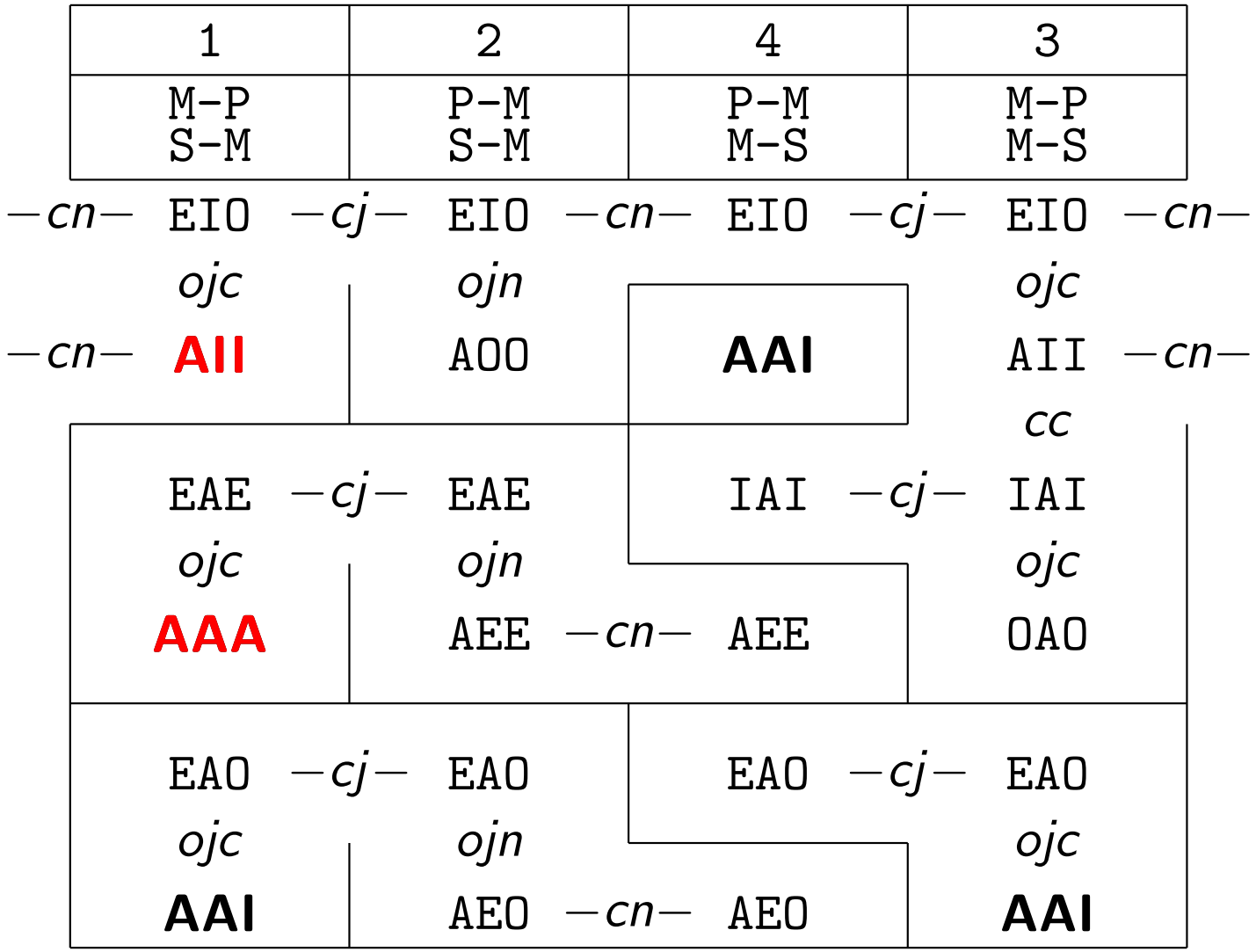
Answer: Subalternation can eliminate axioms A2-A5.
This leaves only Axiom A1, AAA-1, *Barbara*.
cf. $P \vdash P$ as the only axiom of traditional natural deduction.

The 24 valid and conditionally valid syllogisms



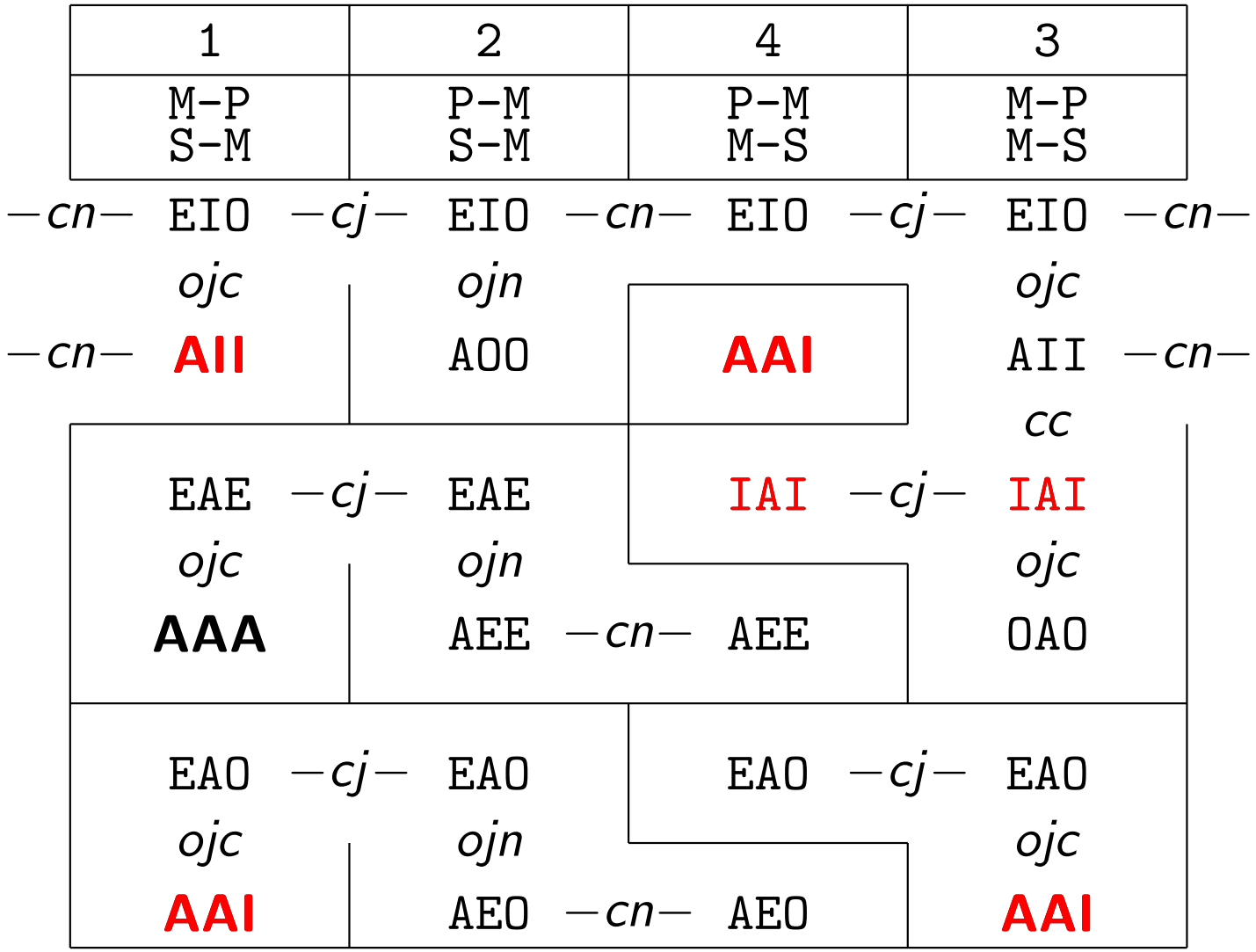
Axioms A1. **AAA-1**, A2. **AII-1**, A3. **AAI-3**, A4. **AAI-4**, A5. **AAI-1**

Eliminating Axiom A2: **AII-1**



Rule R3. In figure 1 or 2, weaken minor premise and conclusion.

Eliminating Axioms **AAI-3, AAI-4, AAI-5**



Rule R4. Conditionally strengthen a particular premise.

Summary of axiomatization

System D4 (following Aristotle's D1/D2 and Corcoran's D3.)

A1. **AAA**-1.

R1. Convert one *e* or *i* form.

R2. Obvert any two sentences with the same RHS.

R3. In figure 1 or 2, weaken the minor premise and the conclusion.

R4. Conditionally strengthen a particular premise.

(R2: In figures 1 and 3, the major premise and the conclusion have the same RHS, namely P, while in figure 2 the two premises have the same RHS, namely M.)

PART 2

BOOLE

The language of Boole

In 1847 George Boole published a pamphlet, *The Mathematical Analysis of Logic*, that proposed a theory of logic consisting of those equations between polynomials that hold identically of the integers, together with the equation $x^2 = x$, namely the condition that multiplication be idempotent.

Today we call this the theory of **Boolean rings**, that is, a ring whose multiplication is idempotent.

In this theory conjunction $x \wedge y$ is realized as multiplication xy and negation $\neg x$ as $1 - x$. By De Morgan's law disjunction $x \vee y$ can be defined as $1 - (1 - x)(1 - y)$, which simplifies to $x + y - xy$.

The polynomials of Zhegalkin

In 1927 Ivan Ivanovich Zhegalkin realized that Boole's theory consisted of those equations between polynomials that hold identically of the integers mod 2.

In this theory subtraction is the same operation as addition and so negation $1 - x$ can be defined as $x + 1$ and similarly disjunction as $x + y + xy$.

A polynomial in this theory is a (finite) sum of square-free monomials without coefficients (i.e. coefficient 1), e.g. $x + y + xy + yz$. 0 and 1 are included, 0 as the empty sum and 1 as the empty monomomial.

These are called Zhegalkin polynomials in Russia, and algebraic normal form or Reed-Muller expansions elsewhere. They are the polynomials of the ring of integers mod 2.

The language of Jevons

Long before Zhegalkin's insight into Boole's language, Stanley Jevons disapproved of Boole's choice of the arithmetic of numbers as a basis for logic and proposed instead the language of conjunction, disjunction, negation, and constants 0 and 1, that we take as standard today.

In this language a Boolean algebra is a complemented distributive lattice.

One benefit of this language is that its two binary operations are mutually dual: $\neg(\neg x \wedge \neg y) = x \vee y$ and dually (De Morgan's laws).

Furthermore negation is self-dual: $\neg\neg\neg x = \neg x$.

Lastly 0 and 1 are mutually dual: $\neg 0 = 1$.

This language therefore has the nice property that the duals of its basic operations are basic operations. That is, the operation basis (set of basic operations) is closed under dualization.

Boole's language of Boolean rings does not have this property, though the language consisting of the Zhegalkin polynomials does.

The language of Heyting

In 1920 Luitzen Egbertus Jan Brouwer's based his concept of intuitionistic set theory on avoidance of the law of excluded middle.

In 1930 Heyting proposed a propositional formalization of intuitionism based on replacing negation in Jevons' language by implication $x \to y$, definable as the weakest proposition making Modus Ponens, $P, P \to Q \vdash Q$, a sound rule of inference.

In this language negation $\neg x$ is definable as $x \to 0$. Boolean algebra is then Heyting algebra together with the law of excluded middle; alternatively with the law of double negation, $(x \to 0) \to 0 = x$.

Heyting's basis of operations is not closed under dualization (the dual of implication is clearly not among Heyting's basic operations).

A dilemma

What is a Boolean algebra?

A Boolean ring?

A complemented distributive lattice?

A Heyting algebra satisfying LEM?

Proposal: the finitary operations on the set $\{0, 1\}$.

Benefits. Neutral. Simple axiomatization. Dual of an operation easily defined. Polynomial-sized proofs (polynomial in number of operations of the theorem to be proved).

Drawbacks: Only locally finite (finitely many operations of a given arity). Huge (albeit finite) operations when written in binary.

MaxBool: A neutral language

MaxBool: the language whose operation symbols are the finitary operations on $\{0, 1\}$. Very similar to but not quite the Zhegalkin polynomials, which treat arity differently.

Each m-ary operation has 2^m possible values of its m inputs, and hence is representable as a bit string of length 2^m whose i-th bit from the right is the value of the operation at the m bits of i.

This represents an operation as its truth table.

Examples: $\neg x_0$: 01. $x_0 \vee x_1$: 1110. $x_0x_1 \vee x_1x_2 \vee x_2x_0$: 11101000.

The difference from Zhegalkin polynomials is that unary x_0 is distinguished from binary x_0 as respectively 10 and 1010, whereas as Zhegalkin polynomials both are x_0.

This distinction partitions the language as the disjoint sum $MB_0 + MB_1 + MB_2 + \ldots$ where each MB_n consists of exactly 2^{2^n} n-ary operations. Zhegalkin polynomials identify some of these.

Duality

Theorem

MaxBool is closed under dualization.

Proof.

The dual of an operation is represented as the complement of the reverse of its representation as a bit string. MaxBool contains all binary strings of length a power of two and hence contains every such dual. □

Examples: $\neg x_0$: 01. $x_0 \vee x_1$: 1110. $x_0x_1 \vee x_1x_2 \vee x_2x_0$: 11101000.

Duals: $\neg x_0$: 01. $x_0 \wedge x_1$: 1000. $x_0x_1 \vee x_1x_2 \vee x_2x_0$: 11101000.

So $\neg x_0$ and $x_0x_1 \vee x_1x_2 \vee x_2x_0$ are self-dual, but $x_0 \wedge x_1$ and $x_0 \vee x_1$ are not.

(The Zhegalkin polynomials also have this property because dualization preserves which arguments an operation depends on.)

Matrix representation

x	1010
y	1100
$x \wedge y$	1000
$x \vee y$	1110
$x \to y$	1101

x	10101010
y	11001100
z	11110000
$x \wedge y$	10001000
$y \wedge z$	11000000
$z \wedge x$	10100000
$(x \wedge y) \vee (y \wedge z) \vee (z \wedge x)$	11101000

A locally finite axiomatization of Boolean algebra

m-ary MaxBool **term**: either an m-ary atom (string of length 2^m or an application $f(t_1, \ldots, t_n)$ where f is an n-ary atom and the t_i's are m-ary terms.

The following system, MB_m, uses the foregoing product to axiomatize T_m, the theory of Boolean algebra based on m variables. The atomic subterms of terms are m-ary operation symbols; all other operation symbols, namely those applied to n-tuples, are of arity n, with no restriction on n.

System $\mathbf{MB}_m$.

A1. $f(t_1, \ldots, t_n) = f \circ T$.

Here f is n-ary and the t_i's are m-ary atoms stacked vertically to form an $n \times 2^m$ bit matrix T.

$f \circ T$ is a form of matrix product, in this case of a 1×2^n matrix with a $n \times 2^m$ matrix, intended as a syntactic realization of application of f to T.

Matrix product

Given a $p \times 2^n$ matrix A and an $n \times 2^m$ matrix B, define their **product** $A \circ B$ to be the $p \times 2^m$ result of replacing each column of B by the column of A indexed by the replaced column. Formally,

$$(A \circ B)_{ij} = A_{it} \quad \text{where} \quad t = \lambda k.B_{kj} = B_{*j} \ (\text{column } j \text{ of } B).$$

Example

A
10101010
11001100
11110000
10001000
11000000
10100000
11101000

$\circ$

B
1000
1110
1101

$=$

$A \circ B$
1000
1110
1101
1000
1100
1000
1100

Completeness

Theorem

MB_m is complete, meaning that it proves every equation $s = t$ of T_m.

Proof.

Case 1: t atomic: Use induction on the height of s.
For height one s is an atom. s is the same term as t if and only if it represents the same m-ary operation.
For height two s must equal a uniquely determined m-ary atom, which $A1$ supplies.
For greater height induction reduces s to height 2 by converting its arguments to atoms.
For general t, if $s = t$ is an identity both must reduce to the same atom a by the foregoing. □

PART 3

CHU, as a path to STONE

Boolean algebras and their homomorphisms

A **homomorphism** of Boolean algebras is a structure-preserving function between them.

Example. $\mathcal{B} = (B, \wedge, \vee, \neg, 0, 1)$, $\mathcal{B}' = (B', \wedge', \vee', \neg', 0', 1')$. A function $h : B \to B'$ is a homomorphism when for all a, b in B, $h(a \wedge b) = h(a) \wedge' h(b)$ and similarly for the other operations. Likewise $h(0) = 0'$, $h(1) = 1'$.

The category **Bool** of Boolean algebras consists of all Boolean algebras and their homomorphisms.

An **ultrafilter** of a Boolean algebra $\mathcal{B}$ is a subset of B whose characteristic function is a homomorphism from $\mathcal{B}$ to the two-element Boolean algebra.

Stone duality

In 1936 Marshall Stone associated to each Boolean algebra $\mathcal{B}$ a totally disconnected compact Hausdorff space S, and showed that $\mathcal{B}$ could be recovered up to isomorphism as the clopen (simultaneously closed and open) subsets of S.

The points of S are taken to be the ultrafilters of $\mathcal{B}$

Furthermore every totally disconnected compact Hausdorff space gives rise to a Boolean algebra in this way.

The category of totally disconnected compact Hausdorff spaces and their continuous functions is called **Stone**.

The category **Bool** is dual to **Stone**, that is, it is equivalent to $\mathbf{Stone}^{\mathrm{op}}$, the opposite of **Stone** obtained by reversing its morphisms.

Chu spaces

We shall explicate Stone duality in terms of transposition of Chu spaces.

A Chu space (A, r, X) over a set K consists of sets A and X and a function $r : A \times X \to K$, that is, an $A \times X$ matrix whose entry at row $a \in A$ and column $x \in X$ is given by $r(a, x)$.

A homomorphism $(f, g) : (A, r, X) \to (B, s, Y)$ of two Chu spaces consists of a pair of functions $f : A \to B$, $g : Y \to X$ satisfying an **adjunction** condition, namely for all $a \in A$ and $y \in Y$, $s(f(a), y) = r(a, g(y))$.

A Chu homomorphism can be understood as a sort of continuous function $f : A \to B$ which preserves structure expressed by the matrix r, with $g : Y \to X$ witnessing the requirement that the inverse image under f of each column of $\mathcal{B}$ is a column of $\mathcal{A}$.

Chu spaces and their homomorphisms form the category $\mathbf{Chu}_K$.

Chu duality

$\mathbf{Chu}_K{}^{\mathrm{op}}$ has a simple representation obtained trivially by transposing every Chu space and switching f and g in every Chu homomorphism.

The category **Bool** is representable as the full subcategory of $\mathbf{Chu}_2$ consisting, for each Boolean algebra $\mathcal{B}$, of those Chu spaces (B, s, Y) for which Y is the set of all ultrafilters of $\mathcal{B}$ and for each element $b \in B$ and each ultrafilter $y \in Y$, $\lambda b.s(b, y)$ is the characteristic function of y.

Example: the 4×2 Chu space

A
11
10
01
00

Conversion to Stone spaces

The crucial property of this representation is that the Chu homomorphisms of this full subcategory of $\mathbf{Chu}_2$ correspond exactly to the homomorphisms of **Bool**. That is, **Bool** embeds fully in $\mathbf{Chu}_2$.

The image of this embedding under transposition is therefore a full subcategory of $Chu_2{}^{\mathrm{op}}$ dual to **Bool**.

Furthermore it is straightforward to see that the set of Chu homomorphisms from $\mathcal{A} = (A, r, X)$ to the 2×1 Chu space whose two rows are 0 and 1 is in a natural bijection with X, since there are X functions $g : 1 \to X$ and each determines a unique $f : A \to 2$. We can then regard the transpose of $\mathcal{A}$ as having those homomorphisms for its points, with the original rows now supplying the columns to make the set of homomorphisms a Chu space.

Simple transposition therefore gives $\mathbf{Bool}^{\mathrm{op}}$.

Stone's theorem

We can now prove Stone's original theorem by closing the columns of the transposed Chu representations to generate Stone's original topology, and showing that this does not change the maps (Chu transforms), which now constitute continuous functions between Stone spaces.

Theorem

The category $\mathbf{Bool}^{\mathrm{op}}$ *obtained by transposing Chu representations of Boolean algebras is equivalent to the category* **Stone**.

Our approch will be to close the columns of the transposed spaces under arbitrary union to yield topological spaces, and then to argue that the Chu homomorphisms are unchanged and now constitute the continuous functions between those topological spaces.

Proof

Proof.

Closing the source columns can only add maps. Let $f : A \to B$ be a function that was not a Chu transform but became one after closing the source. Now the target is still a transposed Boolean algebra so its columns are closed under complement, whence so is the set of their compositions with f. But there are no new clopens in the source, hence no new source column can be responsible for making f a Chu transform, so f must have been a Chu transform before closing the source.

Closing the target columns under arbitary union can only delete maps. But since the new target columns are arbitrary unions of old ones, and all Boolean combinations of columns commute with composition with f, the necessary source columns will also be arbitrary unions of old ones, which exist because we previously so closed the source columns. □