

A CALCULUS OF MATRICAL DESCRIPTORS

DAVID MEREDITH

We present a formal system which, in a particular and limited sense, contains but is not contained in propositional calculus. Section 1 of our paper discusses the motivation underlying the system's development. Section 2 and section 3 present the system itself, and the final section establishes its relationship to propositional calculus. The aim of our paper is simply to arouse interest, and no attempt has been made to treat the subject exhaustively.

1 *What Matrical Descriptors Are* Anyone who has used matrices for the separation of propositional axioms will be aware of some properties of matrices necessary and sufficient for the verification of certain formulae. To verify $C\dot{p}p$, for example, a matrix need merely have designated values on its diagonal. To verify $C\dot{p}Cq\dot{q}$, all that is needed is that the elements \dot{ij} , where j appears on the diagonal, take designated values. Observations like these can be made about formulae in more than a single dyadic functor. The formula $NC\dot{p}p$ will be verified if and only if all the values on the diagonal of the matrix used for the dyadic functor take designated values when they are treated as elements of the matrix used for the monadic functor, and the formula $C\dot{p}NCq\dot{r}$ will be verified if and only if the values of elements \dot{ij} are designated, when j is the value on the monadic matrix for any element that occurs as a value in the dyadic matrix. As the complexity of formulae increases, so does the cumbersomeness of the type of observation we have been making. Propositional formulae are structured so as to interpret readily as statements about propositions, and not as statements about matrices. This latter role is the one that matrical descriptors are designed to play: the system aims to encompass the minimal formal apparatus required to state for any given propositional formula, the necessary and sufficient conditions for a matrix to verify it. A precursor system of the present one was proposed in an RCA Corporation technical report, "The System RCV", distributed by this author in 1966. That system's structure, however, differs fundamentally from the present system's, and constitutes a much less fruitful approach to the problem.

Received November 4, 1972

2 System Morphology

Signs The most elementary expressions in the system are *value variables*

$'a', 'b', \dots, 'k', 'k_1', 'k_2' \dots$

There is a single *undetermined constant value* 'S', an indefinite number of *monadic value operators* ' Γ ', ' Γ_1 ', ' Γ_2 ' . . . and an indefinite number of *dyadic value operators* ' Δ ', ' Δ_1 ', ' Δ_2 ' . . . Other signs used, are signs for arithmetic relations '=', '<=', '>', propositional connectives '.', '⊃', parentheses '(', ')', and quantifiers 'Π', 'Σ'.

Formation Rules The rules governing formation of well-formed formulae are as follows:

Rule 1. Let E be any of the following

- a) A value variable,
- b) The undetermined constant value,
- c) A monadic value operator followed by one value variable,
- d) A dyadic value operator followed by two value variables,

then E is a *value expression*, and the variables in E are free in E .

Rule 2. A *value formula* is defined recursively in three steps.

I Let E_i and E_j be value expressions, and let F be any one of the following

- a) $E_i = E_j$,
- b) $E_i \leq E_j$,
- c) $E_i > E_j$,

then F is a value formula, and the variables free in E_i and E_j are free in F .

II Let F_i and F_j be value formulae, and let F_k be either of the following

- a) $(F_i \cdot F_j)$,
- b) $(F_i \supset F_j)$,

then F_k is a value formula, and the variables free in F_i and F_j are free in F_k .

III Let F_i be a value formula in which α is a free variable, and let F_j be either of the following

- a) $\Pi \alpha F_i$,
- b) $\Sigma \alpha F_i$,

then F_j is a value formula, and α is not free in F_j .

Rule 3. A *matrical descriptor* is defined recursively in two steps.

I Let F be a value formula in which there are no free variables, then F is a matrical descriptor.

II Let D_i and D_j be matrical descriptors, then $D_i \cdot D_j$ is a matrical descriptor.

Conventions To avoid unnecessarily complex expressions we make use of the following conventions:

Convention 1 In a descriptor in which only one dyadic value operator occurs, that operator may be omitted.

Convention 2 In accordance with the rule that ‘ \supset ’ takes precedence over ‘ \cdot ’ and that a string of conjuncts is left associative, parentheses may be omitted where they are not needed to define quantifier scope.

Convention 3 A series of quantifier-variable pairs where the quantifiers are of the same type, may be replaced by a single quantifier followed by the variables.

These conventions enable us to write

$$\Pi ab(\Sigma cde(a = cb \cdot b = cd \cdot c = ed) \supset ab \leq S)$$

in place of

$$\Pi a \Pi b(\Sigma c \Sigma d \Sigma e((a = \Delta cb \cdot b = \Delta cd) \cdot c = \Delta ed) \supset \Delta ab \leq S).$$

Simple Descriptors We focus attention on a particular class of matrical descriptor—simple descriptors—which can be defined with the aid of two auxiliary definitions.

Definition 1 Let L be a value formula of either of the following forms:

- a) $(F_1 \cdot F_2 \dots F_n)$,
- b) $\Sigma \alpha_1 \alpha_2 \dots \alpha_m (F_1 \cdot F_2 \dots F_n)$,

where $n, m \geq 1$ and each F_i ($1 \leq i \leq n$) is of the form $E_j = E_k$ neither of these components being the undetermined constant value; then L is a *simple left value formula*.

Definition 2 Let R be a value formula of the form

$$\phi^n \alpha_1 \dots \alpha_n r S$$

where $n = 1, 2$; ϕ^n is an n -adic value operator; r is either \leq or $>$; and S is the undetermined constant value; then R is a *simple right value formula*.

Definition 3 Let L and R respectively be simple left and right value formulae whose only free variables are $\alpha_1, \dots, \alpha_n$ ($n = 1, 2$); and let M be a matrical descriptor of either of the following forms:

- a) $\Pi \alpha_1 \dots \alpha_n (L \supset R)$,
- b) $\Sigma \alpha_1 \dots \alpha_n (L \cdot R)$,

then M is a *simple descriptor*.

Four types of simple descriptor have the relations of the classic square of opposition, and it is appropriate therefore to name them as follows. (In all four cases: $n = 1, 2$; ϕ^n is an n -adic value operator; and L is a simple left value formula whose only free variables are $\alpha_1, \dots, \alpha_n$).

Definition 4 An A-descriptor is a simple descriptor of the form

$$\Pi \alpha_1 \dots \alpha_n (L \supset \phi^n \alpha_1 \dots \alpha_n \leq S).$$

Definition 5 An I-descriptor is a simple descriptor of the form

$$\Sigma \alpha_1 \dots \alpha_n (L \cdot \phi^n \alpha_1 \dots \alpha_n \leq S).$$

Definition 6 An E-descriptor is a simple descriptor of the form

$$\Pi \alpha_1 \dots \alpha_n (L \supset \phi^n \alpha_1 \dots \alpha_n > S).$$

Definition 7 An O-descriptor is a simple descriptor of the form

$$\Sigma \alpha_1 \dots \alpha_n (L \cdot \phi^n \alpha_1 \dots \alpha_n > S).$$

3 *Intended System Interpretation* For purposes of system application, we assume that matrices are one or two dimensional arrays of integral values ranging from 1 to m , and that a continuous subset of the values, always including 1, is chosen as designated, the highest designated value being less than m . As is customary, we shall speak of the matrix $\mathfrak{M}i$ when $\mathfrak{M}i$ is in fact a set of matrices each of which has m values, and for each of which the same subset of values is designated. The basic relationship between matrices and matrical descriptors is simply defined.

Definition 8 A matrix \mathfrak{M} matches a descriptor D , if and only if when S is taken as the highest designated value of \mathfrak{M} , there is a one to one assignment of value operators in D to functors defined by \mathfrak{M} such that D is true.

We may illustrate this definition by taking the matrix $\mathfrak{M}1$ which has only a single designated value

ϕ^2	1	2	3	4	ϕ^1
*1	1	2	3	3	4
2	1	1	3	3	3
3	1	2	1	1	3
4	1	2	1	1	1

and considering the descriptor

$$D1 \quad \Pi ab(\Sigma c(b = ca) \supset ab \leq S)$$

taking ϕ^2 as Δ . Here the only values of a and b for which a value c can be found to satisfy the descriptor's antecedent are the following:

- $a \ b$
- 1 1 where c may be 1, 2, 3, or 4
- 2 1 where c may only be 2
- 2 2 where c may be 1, 3, or 4
- 3 1 where c may be 3 or 4
- 3 3 where c may be 1 or 2
- 4 1 where c may be 3 or 4
- 4 3 where c may be 1 or 2.

In all these cases, the value of Δab is designated, and $\mathfrak{M}1$ therefore matches D1. Similarly it is easy to see that $\mathfrak{M}1$ matches the descriptor

$$D2 \quad \Pi ab(ab \leq S \cdot a \leq S \supset b \leq S)$$

since with ϕ^2 again taken as Δ , there is no case where Δab and a both take a designated value and b does not. (D2 defines the conditions for a matrix to satisfy the Modus Ponens rule.) Finally we can consider $\mathfrak{M}1$ and the descriptor

$$D3 \quad \Pi ab(\Sigma cd(b = cd \cdot c = \Gamma a) \supset ab \leq S).$$

$\mathfrak{M}1$ does not match this descriptor since with Δ assigned to ϕ^2 and Γ to ϕ^1 the values $a = 1, b = 2, c = 4$, and $d = 2$ —among others—make the antecedent of D3 true, and the consequent false.

In dealing with simple descriptors it is sometimes easier to take advantage of their form and simply consider all combinations of values for their value variables, rather than examine each element of the matrix in turn. Thus, for example, given the matrix $\mathfrak{M}2$

$\mathfrak{M}2$	ϕ^2	1	2
	*1	1	2
	2	1	1

and the descriptor

$$D4 \quad \Pi ab(\Sigma c(b = cc \cdot b = ca) \supset ab \leq S)$$

we might proceed in this manner to show that $\mathfrak{M}2$ matches D4.

	1	2	3	4	5
$a b c$	$b = cc$	$b = ca$	$ab \leq S$	1,2	$4 \supset 3$
1 1 1	1 = 11	1 = 11	11 ≤ 1	T	T
1 1 2	1 = 22	1 = 21	11 ≤ 1	T	T
1 2 1	2 = 11	2 = 11	12 ≤ 1	F	T
1 2 2	2 = 22	2 = 21	12 ≤ 1	F	T
2 1 1	1 = 11	1 = 12	21 ≤ 1	F	T
2 1 2	1 = 22	1 = 22	21 ≤ 1	T	T
2 2 1	2 = 11	2 = 12	22 ≤ 1	F	T
2 2 2	2 = 22	2 = 22	22 ≤ 1	F	T

4 Matrical Descriptors and Propositional Formulae We define a relationship of correspondence between matrical descriptors and propositional formulae.

Definition 9 Let D be a matrical descriptor and P a propositional formula, then D and P correspond if and only if for all matrices \mathfrak{M} , \mathfrak{M} matches D if and only if \mathfrak{M} verifies P.

On the basis of this relationship the two following theorems concerning propositional calculus and the calculus of matrical descriptors can be asserted.

Theorem 1 For every propositional formula P , an A-descriptor D can be found such that P and D correspond.

Theorem 2 There exist A-descriptors which have no corresponding propositional formulae.

In proof of the first of these theorems we will describe a procedure for obtaining A-descriptors from propositional formulae and show that the A-descriptor must correspond to the formula from which it was obtained. In proof of the second theorem we will instance some A-descriptors for which corresponding propositional formulae cannot be constructed.

In describing our procedure for obtaining A-descriptors from propositional formulae, we utilize the following meta-notation.

' ϕ^n ' ($n = 1, 2$) is an n -adic propositional functor.

' α ', ' β ' denote propositional formulae or non-elementary expressions formed from propositional formulae by the replacement of well-formed components by value variables from the calculus of matrical descriptors.

' x ', ' y ', ' z ', ' z_1 ', ' z_2 ' ... are value variables of the calculus of matrical descriptors.

' E_1 ', ' E_2 ', ... are expressions of the form $x = \alpha$.

' $[f/g]h$ ' denotes the expression obtained from h by replacing every occurrence of f with g .

'*' denotes a substitution operation defined as follows:

- a) If α and β are independent, * is $\alpha/z_1, \beta/z_2$.
- b) If β occurs in α , * is $\alpha/z_1, [\beta/z_2]\alpha/z_1, \beta/z_2$.
- c) If α occurs in β , * is $\alpha/z_1, \beta/z_2, [\alpha/z_1]\beta/z_2$.
- d) If α and β are identical, * is α/z_1 .

Our procedure involves application of the following six rules. After applying whichever of Rule 1 or Rule 2 is appropriate, Rules 3 and 4 are applied repeatedly until they can be applied no more. A single application is then made of either Rule 5 or Rule 6.

Rule 1 If the given formula is $\phi^1\alpha$ (ϕ^1 null for the assertion function) replace it by

$$F \supset \Gamma\alpha \leq S$$

where F is:

- a) $a = \alpha$ if α is elementary,
- b) $a = \alpha$ if α is non-elementary;

Γ being in both cases the value operator assigned to ϕ^1 .

Rule 2 If the given formula is $\phi^2\alpha\beta$ replace it by

$$F \supset \Delta ab \leq S$$

where F is:

- a) $a = [\beta/b]\alpha \cdot b = [\alpha/a]\beta$ if α and β are distinct and neither is elementary,
- b) $a = \alpha$ if α and β are distinct and both are elementary,
- c) $a = \alpha \cdot b = \alpha$ if α and β are identical and non-elementary,

- d) $a = b$ if α and β are identical and elementary,
- e) $b = [\alpha/a]\beta$ if α is elementary and β is not,
- f) $a = [\beta/b]\alpha$ if β is elementary and α is not;

Δ being in all cases the value operator assigned to ϕ^2 .

Rule 3 In any antecedent ending with

$$x = \phi^1\alpha \text{ or } x = \phi^1\alpha \cdot E_1 \dots E_n$$

and not containing z_1 , replace these expressions by respectively

- a) $x = \Gamma z_1 \cdot z_1 = \alpha$ and $x = \Gamma z_1 \cdot z_1 = \alpha \cdot *E_1 \dots *E_n$ if α is non-elementary,
- b) $x = \Gamma z_1$ and $x = \Gamma z_1 \cdot *E_1 \dots *E_n$ if α is a propositional variable;

and in any antecedent ending with

$$x = \phi^1y \text{ or } x = \phi^1y \cdot E_1 \dots E_n$$

replace these expressions by respectively

- c) $x = \Gamma y$ and $x = \Gamma y \cdot E_1 \dots E_n$;

Γ being in all cases the value operator assigned to ϕ^1 .

Rule 4 In any antecedent ending with

$$x = \phi^2\alpha\beta \text{ or } x = \phi^2\alpha\beta \cdot E_1 \dots E_n$$

and not containing z_1 or z_2 , replace these expressions by respectively

- a) $x = \Delta z_1 z_2 \cdot z_1 = [\beta/z_2]\alpha \cdot z_2 = [\alpha/z_1]\beta$ and $x = \Delta z_1 z_2 \cdot z_1 = [\beta/z_2]\alpha \cdot z_2 = [\alpha/z_1]\beta \cdot *E_1 \dots *E_n$ if α and β are distinct and neither is elementary,
- b) $x = \Delta z_1 z_2$ and $x = \Delta z_1 z_2 \cdot *E_1 \dots *E_n$ if α and β are distinct and both are propositional variables,
- c) $x = \Delta z_1 z_1 \cdot z_1 = \alpha$ and $x = \Delta z_1 z_1 \cdot z_1 = \alpha \cdot *E_1 \dots *E_n$ if α and β are identical and non-elementary,
- d) $x = \Delta z_1 z_1$ and $x = \Delta z_1 z_1 \cdot *E_1 \dots *E_n$ if α and β are identical propositional variables,
- e) $x = \Delta z_1 z_2 \cdot z_2 = [\alpha/z_1]\beta$ and $x = \Delta z_1 z_2 \cdot z_2 = [\alpha/z_1]\beta \cdot *E_1 \dots *E_n$ if α is a propositional variable and β is non-elementary,
- f) $x = \Delta z_1 z_2 \cdot z_1 = [\beta/z_2]\alpha$ and $x = \Delta z_1 z_2 \cdot z_2 = [\beta/z_2]\alpha \cdot *E_1 \dots *E_n$ if β is a propositional variable and α is non-elementary;

and in any antecedent ending with

$$x = \phi^2\alpha y \text{ or } x = \phi^2\alpha y \cdot E_1 \dots E_n$$

and not containing z_1 , replace these expressions by respectively

- g) $x = \Delta z_1 y \cdot z_1 = \alpha$ and $x = \Delta z_1 y \cdot z_1 = \alpha \cdot *E_1 \dots *E_n$ if α is non-elementary,
- h) $x = \Delta z_1 y$ and $x = \Delta z_1 y \cdot *E_1 \dots *E_n$ if α is a propositional variable;

and in any antecedent ending with

$$x = \phi^2y\alpha \text{ or } x = \phi^2y\alpha \cdot E_1 \dots E_n$$

and not containing z_1 , replace these expressions by respectively

- i) $x = \Delta yz_1 \cdot z_1 = \alpha$ and $x = \Delta yz_1 \cdot z_1 = \alpha \cdot *E_1, \dots, *E_n$ if α is non-elementary,
 j) $x = \Delta yz_1$ and $x = \Delta yz_1 \cdot *E_1, \dots, *E_n$ if α is a propositional variable;

and in any antecedent ending with

$$x = \phi^2 yz \text{ or } x = \phi^2 yz \cdot E_1, \dots, E_n$$

replace these expressions by respectively

- k) $x = \Delta yz$ and $x = \Delta yz \cdot E_1, \dots, E_n$;

Δ being in all cases the value operator assigned to ϕ^2 .

Rule 5 Given an expression

$$E_1, \dots, E_n \supset \Gamma a \leq S$$

replace it by

- a) $\Pi a(E_1, \dots, E_n \supset \Gamma a \leq S)$ if a is the only variable and is free,
 b) $\Pi a(\Sigma z_1 \dots z_m(E_1, \dots, E_n) \supset \Gamma a \leq S)$ if a, z_1, \dots, z_m are the only variables and all are free.

Rule 6 Given an expression

$$E_1, \dots, E_n \supset \Delta ab \leq S$$

replace it by

- a) $\Pi ab(E_1, \dots, E_n \supset \Delta ab \leq S)$ if a and b are the only variables and both are free,
 b) $\Pi ab(\Sigma z_1 \dots z_m(E_1, \dots, E_n) \supset \Delta ab \leq S)$ if a, b, z_1, \dots, z_m are the only variables and all are free.

In the following illustrations of this procedure Δ is the value operator assigned to C , Δ_1 the operator assigned to K , and Γ the operator assigned to N . The rule used to obtain each line from its predecessor is noted to its right.

Ex. 1

$$CC\bar{p}C\bar{p}qC\bar{p}q$$

$$a = C\bar{p}b \cdot b = C\bar{p}q \supset \Delta ab \leq S \quad (2a)$$

$$a = \Delta cb \cdot b = Cc\bar{q} \supset \Delta ab \leq S \quad (4h)$$

$$a = \Delta cb \cdot b = \Delta cd \supset \Delta ab \leq S \quad (4j)$$

$$\Pi ab(\Sigma cd(a = cb \cdot b = cd) \supset ab \leq S) \quad (6b)$$

Ex. 2

$$CCN\bar{p}\bar{p}\bar{p}$$

$$a = CNbb \supset \Delta ab \leq S \quad (2f)$$

$$a = \Delta cb \cdot c = N\bar{b} \supset \Delta ab \leq S \quad (4g)$$

$$a = \Delta cb \cdot c = \Gamma b \supset \Delta ab \leq S \quad (3c)$$

$$\Pi ab(\Sigma c(a = cb \cdot c = \Gamma b) \supset ab \leq S) \quad (6b)$$

Ex. 3

$$CK\bar{p}q\bar{p}$$

$$a = K\bar{b}q \supset \Delta ab \leq S \quad (2f)$$

$$a = \Delta_1 bc \supset \Delta ab \leq S \quad (4j)$$

$$\Pi ab(\Sigma c(a = \Delta_1 bc) \supset \Delta ab \leq S) \quad (6b)$$

It is not difficult to establish that our procedure will always lead to an A-descriptor which corresponds to the formula from which it was obtained. Rules 1 and 2 assign names to the arguments of the formula's major functor, preserving the relationship between these arguments in the descriptor's antecedent, and providing an always true antecedent for formulae with a single monadic functor and one variable or with a single dyadic functor and two distinct variables. Rules 3 and 4 continue the breakdown of complex arguments, ensuring that the same argument, wherever it occurs, receives the same name, and that distinct arguments receive distinct names. The last two rules merely provide the appropriate quantifiers. The relationship between the arguments of all functors in the original formula is thus preserved in the descriptor's antecedent, and since value operators can always be so chosen that there is a one to one correspondence between them and functors of the formula, this antecedent will be true if and only if there is an assignment of values to the variables in the propositional formula that gives the whole formula the value of the major functor for i in case this is monadic and for ij in case this is dyadic. The descriptor's consequent is true, however, if and only if this value is designated, and correspondence is thus assured.

As an instance of an A-descriptor with no propositional correspondent we can cite

$$D5 \quad \Pi ab(a = ab \supset ab \leq S).$$

A matrix matches this descriptor if and only if the value of $\phi^2 ab$ is designated whenever it is identical with a . For a propositional formula to be verified by all matrices satisfying this condition, the whole formula would have to be identical with the first argument, which is impossible. The descriptor D4 given in the previous section is also one with no propositional correspondent, since we cannot construct a formula whose value will be designated whenever the value of the second argument occurs both as the value of any element ij , where j is the value of the first argument, and as the value of the diagonal element ii . (Note that $CpC\phi\phi$ corresponds, not to D4, but to its subinstance below.)

$$D6 \quad \Pi ab(b = aa \supset ab \leq S).$$

Information International, Inc.
Culver City, California