My first referee's report
as an academic.
(took two days)

2. Sept 98.

Rudolph Bayer.

22nd July, 1969

J. C. Reynolds, Esq.,
Applied Mathematics Division,
Argonne National Laboratories,
Argonne, Illinois, 60439,
U.S.A.

Dear John,

I enclose a referee's report on the enclosed paper, as promised.

It seems a good and worthwhile paper, tackling a subject of some importance. I recommend publication.

My main suggestions for change are clarifications, stylistic changes, and the removal of minor inaccuracies. One or two lacunae in the exposition have been pointed out, and should be filled. I think the incorporation of my suggestions will improve readability, but I would not insist on them in their entirety.

I still think the paper goes further than either the reader or the practical implementor is likely to follow. The authors, although they refer to the possibility (or even the ease) of implementation seem to have little idea what an elaborate structure they have set up, and cannot therefore understand how pursuit of the ultimate generality may make satisfactory implementation impossible - a common failing of theorists who have had no implementation experience.

I'm afraid I have omitted the detailed checking of the technical material between page 51 and page 63 inclusive. I just haven't got the time at the moment. If you can't get another referee to do it, let me know, and I'll have another bash!

I still haven't got the proofs of my own paper, though promised.

Yours sincerely,

GENERAL

1. I would suggest a change to the title of the paper, to express the importance of its subject matter more clearly. Perhaps:

   "Matrix calculi and their optimisation for efficient computation"

   There is no need to introduce the "puff" words "Data Structure"

   Furthermore it is misleading to entitle the paper "complete" calculi, when the recommended calculus is incomplete.

2. In general the paper makes the practice of introducing concepts and notations long before they are needed. For example sequences and extensions are introduced on pages 4 and 5, but are not needed until page 47. It would certainly produce a more readable paper if introduction of the more advanced concepts and notations could be postponed until they are needed.

   The interest and importance of the paper, and the general confidence of the reader, would be much increased if the authors were able to include a few examples which have occurred in "real life" together with their normal forms; since this would give a chance to evaluate the practical impact of the theoretical studies.

   Finally, though it is probably too late now, it is really quite a good idea to give some sort of rigorous syntactic definition of the class of expressions which are to be considered in each calculus. Without such a definition, it is rather meaningless to quote rigorous theorems on the completeness of the calculi, and rather cruel to ask the reader to convince himself of their validity.

DETAILS

Page 1. two lines from bottom page 2.

for "Then similarly as in [4]"

read "Then as in [4],"

reason style

page 3 after first sentence insert "The operators $\triangleleft$ and $\triangleright$ are known as extraction operators"

reason to introduce the technical term used on page 20

2. page 4 lines 4 and 5

remove the inverted commas

reason unnecessary

3. page 4 middle

for "Concatenation." read "The Concatenation . . . "

for "defined" read "obtained"

reason stylistic accuracy

4. page 4 bottom

run the last three paragraphs into one.

5. page 5 six lines from bottom

for "given operation w..." (to bottom of page)"

read "given operation w. This is known as the cumulative operator arising from w, and will be written

$$\int w \cdot A = ((\ldots (A_1 w A_2) w \ldots) w A_e)$$

We prefer the rotation $\int w$ to the form w/ introduced by Inverson[4]. $\int w$ is

reason emphasise finitude of sequence

eliminate double reference to Inverson.

clarify the definition of "cumulative"

6. Introduce broken lines and dots into the diagram to make it clear that it is not shown in full.

Page 7. line 10

after necessary

insert , in the interests of speed or storage economy,

reason to clarify the motivation

7. line 16

for "in matrix concatenations which then . . . "

read ", freely using the matrix operations, including concatenation, as described above. These expressions are then . . ."

7. bottom paragraph

8. top paragraph

omit these two paragraphs

reason they tend to confuse the main issue at this stage

8. line 11

omit "One must . . . room for interpretation"

reason the text is shorter and more straightforward without these two sentences. The point which they make is philosophical, and if introduced cannot be treated so lightly.

8. two lines from bottom

do not split the paragraph here

9. line 9

new paragraph at "Similar remarks . . . "

9. second line from bottom

for "one writes"

read "the original assignment were to be transformed into"

reason emphasises that the objective is to transform the expressions automatically, not to get the user to write them more efficiently

10. line 6

for "by assignments"

read "by sequences of assignments"

reason more strictly accurate

Page 10. line 12

for "The range R specifies"

read "The range R in general specifies"

reason accuracy

11. line 1

after "such normal forms"

insert "as will promote efficiency of calculation."

11. no new paragraph at "A similar situation . . . "

or at "A further complication . . . "

or at "We call a . . . "

12. I would suggest that the defined words should be taken into the text rather than displayed so emphatically.

12. seven lines from bottom

for an expression

read any well-formed expression

13. line 1-2

for "one can be transformed into the other"

read "if they can both be transformed to the same normal form"

reason accuracy, in view of the fact that the transformation rules are directional.

13. line 5

for "defining the transformation rule . . . (to end of paragraph)"

read "may have a meaningful value, and the other may not. This is because our transformation rules may sometimes transform a meaningless expression into a meaningful one. However, since our transformation rules operate in one direction only, they will never transform a meaningful expression into a meaningless one. Thus whenever . . ."

Page 13. line 12

after specify

insert in advance

Page 13. line 14

for "The bulk . . . subsequent developments"

read This set may be regarded as being defined as the set of expressions which are formed by the normal method of composition of binary and unary operators, possibly omitting brackets in accordance with stated precedence rules. Of course, in the design of an actual language based on the theory expounded here, it would be necessary to give more strict syntactic definitions of the well-formed expressions, using Backus Normal Form or some similar technique"

13. for "However, we . . ."

read "We . . ."

13. do not display "normal form"

14. Omit paragraph beginning "Our not formally . . ."

reason avoid unnecessarily controversial statements, avoid dealing with same (trivial) problem in two separated paragraphs. Later on, i.e. bottom of page 24, it is assumed that the reader will know exactly the class of well-formed expressions.

14. line 5

replace paragraph beginning "Since every . . ."

by It is important to note that the term "expression" here is used in a broader sense than in programming languages: it includes complete assignment statements, and even sequences of assignments. Since every assignment statement contains := as its outermost operator, any assignment statement (or sequence of assignments) in normal form must be free of concatenation operator.

reason (a) the original paragraph is not quite accurate in the case of sequences of assignment statements.

(b) it is necessary to clarify the use of the term "expression" in this paper, since it is slightly wider than normal.

Page 15. no new paragraph on second to last line.

15. line 15

insert "However, we assume that the extraction operators < and > may appear on the left of :=, together with their associated range expression."

16. add a new operator: $\mathcal{L}$) statement composition.

As in ALGOL 60, we use the semicolon (;) to separate assignment statements which are to be executed in sequence. Like := this is not a strict operator, since its "operands" can only be assignment statements, (or sequences thereof), but for purposes of symbolic manipulation it behaves much like one.

17. the shift operator σ

You have already used σ for the concatenations of lists. The ambiguity should be removed preferably by changing one of the symbols. How about using the PL/I notations I4/K for concatenation?

18. line 7

for minus 1

read, them finally add 1.

reason clarity

18. line 11

for first occurrence of $\curvearrowright$

read $\curvearrowright$

remove primes from Q and R

reason confusing and unnecessary

18. line 14

for "very simple calculus"

read "a simple matrix calculus which omits the concept of a sequence and its

associated operators, and in which all ranges are assumed to be monotonic."

reason the calculus presents very considerable complexity for an implementor. It seems therefore misleading to call it "very simple".

18. four lines from bottom

add ; to list of matrix operators.

Page 19. line 3

the o should be small raised circle °

add ; to the end of the list

Note: in general it is better to allow certain operators to have the same priority, with a defined association (say from the left).

Thus the priority might be defined: extraction operators, multiplication operators, adding operators, concatenation operators, assignment, ## range operators.

Page 20. line 11

move paragraph beginning "The identities . . ." to the end of R1 on the next page

22. after line 4

insert a rule for A↘B := C

22. line 5

replace "We use . . . other expressions"

by "We use ZEROS to stand for a matrix consisting wholly of zeros; its dimensions can readily be determined by context.

reason (a) Earlier suggested changes have already introduced the ; operator in a satisfactory fashion

(b) some explanation of the term ZEROS is in order.

Page 22 line 10

omit "Let R . . . arbitrary range"

Page 23 line

for "necessary that. . is monotonic"

read "sufficient that R be monotonic. However, Q in the above rules could be an arbitrary range if desired."

reason we have already admitted that the calculus as a whole is complete only for monotonic ranges.

Page 23 after line 11

insert Similar rules apply to - x and /.

Page 24 7 lines from bottom

for "imply" read "simply"

Page 24 4 lines from the bottom

omit "easily"

reason you can put it back if you have done it, and found it easy!

Page 25 line 6

for for complete read complete with respect to the normal form defined in this section

reason completeness is defined only in respect of a given normal form.

Page 25 -26

the phrase "by Rn" should be right justified on each occasion of its occurrence

Page 25 four lines from bottom

insert @ before E

Page 24 for "We then have:"

read Furthermore, if the original expression was an assignment or sequence of assignments, the normal form will not contain any concatenation operator. We then have:

reason it seems worth while to point out this useful fact about normal forms.

Page 28 The author may wish to contrast the code which would result from a direct translation of the original expression:

```
for i: = | step | until rows (A) do
for j: = | step | until columns (A) do
begin S: = 0;
    for k: = |step | until rows (B) + rows (D) do
    S: = S + (if i ≤ rows (B)∧j≤columns (B) then B[i,j] + C[i,j]
    else if rows (B)<i∧columns (B)<j then D[i,j] else 0)
    X (if j≤ rows (E) then E[i,j] else F[i,j]);
    A[i,j]: = S
    end
```

Page 30 line 1

insert missing subscript w after LD*

Page 29 lines 2-4

There is no need to put dashes on the x's and y's

Page 31 lines 8-11

Propositions 6.3 and 6.4 can be combined by the technique of the Note i.e. "symmetric (standard)"

Page 32 line 8

(C4) doesn't exist. You will need to insert some remarks on many operators and their extensions, probably at the end of page 30.

Page 33 One should choose a new symbol for concatenation or for shifting ranges.

Page 34 line 4

for "one"

read "a single"

reason to introduce the technical term "single extension"

Page 35 lines 2,3, and 4

for "and single extensions"

read "and their single extensions"

Page 35 line 14

for identities

read

35. line 15

omit the subtitle "Boosting Equalities"

reason it is unnecessary and confusing.

36. line 14

It's hardly necessary to insert ($\forall_c$), only to remove it again in step 3!

We should deal at the bottom of page 35 with postfix monadic operators ν, say 5) $V[i] \equiv (V^{\nu})[i]$

At least the problem of postfix # should be mentioned, even if it is solved only be an appeal to the good will of the reader.

36. four lines from end

It is far from clear what the interpretation of these rules willbe in the case of identities containing := and ;. I think some special discussion should be given to these cases and the interpretation of their extensions. (e.g. *; *;*)

35. before first line

insert " 0) ;"

reason in many normal forms ; is the outermost operator.

19. before line 9

insert 0) ;

reason same as above

38. line 5

You should also explain how to choose the variables with respect to which the rule should be boosted.

38. line 6

no new paragraph at "Finding . . ."

38. line 12

for simplification rules

read reduction rules

reason the word "simplification", here and elsewhere is likely to provoke a

Page 38. line 12 (cont.)

hollow laugh.

38. lines 13-14

first two lines of this rule on a single line

38. line 11

for binary operator

read binary operator other than :=

38. line 15

insert brackets round ($S \sigma T$)

38. after line 20

insert Note the right extension :=* is meaningless. The meaning of the other extensions of := may be regarded as defined by the reduction rules given above.

39. line 6-7 and 9-10

omit also, e.g. . . . operator

reason surely, if w, are extensions already, $*w*$ and $\mu*$ will go beyond the limits of our present calculus.

40. line 1

read is a complete calculus with respect to the normal form defined in this section.

43. last line

for "expression is the"

read "expression is either in normal form, or is the"

44. at end

it may be kind to insert a remark:

"The proviso that both operators are of height > 0 ensures that the process of deflation, which involves distributing a subscript, will never lead to the subscripting of an expression which is not a sequence"

Page 43. somewhere

It may be worthwhile to insert a remark somehwere "We shall be using the fact that the only operator of height 0 which may be validly applied to a sequence are σ, Δ and the unary operator λ".

41. throughout:

for "ranges"

read "monotonic ranges"

41.

The "operator" ; should not be omitted

42. before line 4

insert " 0) ;"

43. lines 12-14

for ; read ,

The use of ; in these lists may be confused with ;'s occurring in the expressions themselves.

43. five lines from bottom

for simplify

read reduce

44. bottom and second to bottom line

for simplification

read reduction

47. fourth line from bottom

read is a complete calculus with respect to the normal form defined in this section

47. end of section 8.

The authors may care to insert a paragraph to the effect:

"In view of the comlexity of the deflating and boosting algorithms, and the time and space involved in executing them, a practical implementation of a matrix calculus may settle for the single-extension calculus described in section 5.

Page 49. four lines from bottom

Is it worth noting that can be applied only to sequences of matrix?

50. line 9

add at the end of the sentence,

", unless both ranges are null."

50. before line 12

add line " 0);"

52. three lines from bottom

for analogue

read analogous

54.& 57. at several places

for simplification

read reduction

54. top

another alternative is to extend the class of normal forms.

54. line 6

for principal

read principle

54. line 9

omit "at this time".

reason is this judgement likely to vary with time?

57. The problem of the non-uniqueness of the normal form cannot really be dismissed so lightly. The use of the work "complete" in connection with a calculus is justified only if there is a decision procedure (e.g. reduction to the same normal form) which can be mechanically applied to decide the equivalence of any two well-formed expressions of the calculus. If an expression can be reduced to two different normal forms, it is no longer possible to establish even the equivalence of an expression with itself!

Page 57. (cont.)

The easiest way of "curing" this problem is simply to omit all reference to completeness, and to emphasise that the purpose of reduction is to optimise an expression for computation, rather than to solve the decision problem for equivalence.

58. do not display the defined terms "permutation range" and "identity range"

64. line 2

I think reference should also be made to GR3

60. line 6

for "conjecture"

read "conjecture, which turns out to be justified,"

reason If the last paragraph of page 63 is valid, it seems important to avoid discouraging the reader at this stage with the thought that he is only following up a conjecture.