NordDATA90
GÖTEBORG 11-14 JUNI 1990

ORDINAL FRACTIONS - THE ALGEBRA OF DATA

Bo Jacoby, edp-consultant, COWIconsult, Copenhagen.
(Electronic edition, November 2000)

The theory of Ordinal Fractions evolved between 1980 and 1983, but was scarcely recognized. It was said: "It's a fine theory, but seven years ahead of time". Now the seven years have passed. Is time ripe to value simple solutions to complicated problems?

1. **Abstract**

The concept, notation, algebra and relations of ordinal fractions are introduced. Representing subsets, hierarchies, arrays and networks in the same way, ordinal fractions unifies various data-base approaches. It is applicable for knowledge representation and artificial intelligence, because the algebraic properties of of ordinal fractions reflect the logical properties of natural language. The conventional approach to computer science using variable names is shown to be analogous to the use of Roman numbers.

2. **Introduction**

While the theory of fractions is tought in school, the theory of ordinal fractions is new. Integers are cardinal numbers, like 3 or 8, used for counting a set of objects, or ordinal numbers, like the 3'rd or the 8'th, used for addressing objects within a set. Similarily, fractions are cardinal fractions, like 1/2 of 3/8, used for measuring, and ordinal fractions, like the 1'st half or the 3'rd 8'th, used for identifying a part of something.

<u>EXAMPLES</u>	
CARDINAL NUMBER	three
CARDINAL FRACTION	three eighths
ORDINAL NUMBER	the third
ORDINAL FRACTION	the third eighth

3. **Notation**

3/8 is in binary notation 0110, meaning 0/2 + 1/4 + 1/8 + 0/16. Similarily, the 3'rd 8'th is written 1210, meaning 1'st half, 2'nd 4'th, 1'st 8'th, both 16'ths. It is convenient to fill unused digitpositions with zeroes, writing 1210 instead of 121. Digit zero means: no restriction on the value of this digitposition.

<u>BINARY NOTATION</u>	
three	0011
three eighths	0110
the third	0011
the third eighth	1210

4. Let's see how this notation is used to address **subsets, hierarchies, arrays**, and **networks.**

Subsets

The ordinal fraction addresses subsets in a compact and systematic way like this:

```
000  xxxxxxxx  the whole
100  xxxx----  the first half
200  ----xxxx  the second half
010  xx--xx--  the odd fourths
110  xx------  the first fourth
210  ----xx--  the third fourth
020  --xx--xx  the even fourths
120  --xx----  the second fourth
220  ------xx  the fourth fourth
001  x-x-x-x-  the odd eighths
101  x-x-----  the first and third eighths
201  ----x-x-  the fifth and seventh eighths
011  x---x---  the first and fifth eighths
111  x-------  the first eighth
211  ----x---  the fifth eighth
021  --x---x-  the third and seventh eighths
121  --x-----  the third eighth
221  ------x-  the seventh eighth
002  -x-x-x-x  the even eighths
102  -x-x----  the second and fourth eighths
202  -----x-x  the sixth and eighth eighths
012  -x---x--  the second and sixth eighths
112  -x------  the second eighth
212  -----x--  the sixth eighth
022  ---x---x  the fourth and eighth eighths
122  ---x----  the fourth eighth
222  -------x  the eighth eighth
```

Not all sets of eighths are ordinal fractions. There are 256 different sets of eighths, only 27 of which are binary ordinal fractions.

Hierarchies

The nodes of a 3-level hierarchy are numbered like this.

```
000
  100
    110
      111
      112
    120
      121
      122
  200
    210
      211
      212
    220
      221
      222
```

Arrays

Consider a 2-times-2 array

11	12
21	22

The first digit is the row-index and the second digit is the column-index. The rows are `10` and `20` and the columns are `01` and `02.` The complete array is `00.`

00	**01**	**02**
10	11	12
20	21	22

Having got ten objects to number, don't count `1 2 3 4 5 6 7 8 9 10` because this does not recognize the special meaning of digit zero. Don't count `0 1 2 3 4 5 6 7 8 9` or `1 2 3 4 5 6 7 8 9 0` but:

11	12	13	14	15
21	22	23	24	25

Zero is not considered a number, because you don't count to zero. You need two digits for numbering ten objects, because there are only nine one-digit numbers. Ten objects chould be arranged in rows and numbered by row- and column-numbers. Nothing is lost by restricting array-indices to be one-digit-numbers, making delimiters between them superfluous.

Networks

Like arrays and hierarchies, networks may also be represented by means of ordinal fractions. The boxes might be activities of a project, electronical or logical components etc. The point is that they are arranged alternatively in series and in parallel.

In the diagram below, `1000` consists of the leftmost 6 boxes which are connectet in series to box `2000`. `1000` consists of `1100` in parallel with `1200`. `1100` is `1110` in series with `1120`. `1120` is `1121` in parallel with `1122`. `1200` is `1210` in series with `1220` and `1230`.

```
              -1121-
    --1110-|        |-------
   |          -1122-        |
---|                        |--2000
   |                        |
    --1210---1220---1230--
```

The table below contains exactly the same information as the seemingly more complicated diagram above. Either can be constructed from the other.

1110
1121
1122
1210
1220
1230
2000

5. **Algebra**

An ordinal fraction can be regarded a *logical proposition* or a *set*.
For example, the logical proposition that the variable `I` assumes the value `1` , `(I=1)`, or the set of objects `x` for which the proposition is true, `{x : I(x)=1}`, both match the ordinal fraction: `1`

The *conjunction* of logical propositions:

```
(I=1) AND TRUE <=> (I=1)
```

or the *intersection* of sets:

```
{x : I(x)=1} INTERSECTION {EVERYTHING}
= {x : I(x)=1}
```

both match the *addition* of ordinal fractions:

```
1 + 0 = 1 .
```

The addition-table of one-digit binary ordinal fractions is this:

+	0	1	2
0	0	1	2
1	1	1	Ø
2	2	Ø	2

where Ø is the *improper* ordinal fraction, matching the *false* logical proposition or the *empty* set.

If `i` and `j` are different nonzero digits, the formulas for addition are

```
0 + 0 = 0
i + 0 = 0 + i = i + i = i
i + j = Ø
```

Multifigure ordinal fractions are added digit by digit without carry, but if any resulting digit is improper, the whole result is improper. For example:

```
00 + 21 = 21
20 + 01 = 21
21 + 21 = 21
21 + 22 = Ø
```

Like decimal fractions, ordinal fractions are left justified and padded with zeroes before they are added:

```
2 + 03 + 0001 = 2000 + 0300 + 0001 = 2301
```

6. **Relations**

Ordinal fractions `A` and `B` may be compared to one another. There are 8 possible results, called `A::B`, defined by this piece of code:

```
A::B  :=
     if (A=B=Ø) then 8
else if (A=  Ø) then 7
else if (  B=Ø) then 6
else if (A=B  ) then 5
else if (A+B=A) then 4
else if (A+B=B) then 3
else if (A+B=Ø) then 2
else                 1
```

Based on this comparison the following *relations* between ordinal fractions are specified.

`A::B=4,7`	`A<B`	strict subordination
`A::B=4,5,7,8`	`A≤B`	subordination
`A::B=5,8`	`A=B`	equality
`A::B=3,5,6,8`	`A≥B`	superordination
`A::B=3,6`	`A>B`	strict superordination
`A::B=2,6,7,8`	`A><B`	incompatibility
`A::B=1`	`A<>B`	coordination
`A::B=1,3,4,5`	`A≤≥B`	compatibility

Incompatibility `(A><B)` means that `A+B=Ø`. They have *nothing* in common. `10><21`.
Compatibility `(A≤≥B)` means that `A+B>Ø`. They have *something* common. `10≤≥02`, `10≤≥12`, `12≤≥10`.
Subordination `(A≤B)` means that `A+B=A`. `10≤00`.
Superordination `(A≥B)` means that `A+B=B`. `10≥11`.
Coordination `(A<>B)` means that they are compatible without one being subordinate to the other. `10<>02`.

7. **Knowledge representation**

The following example contains words labelled by ordinal fractions.

```
0001 girl
0002 boy
001 nice
002 naughty
01 blackhaired
02 blonde
03 redhaired
1 big
2 little
```

Here the equation $2301=2+03+0001$ matches the sentence: ‘a little redhaired girl is little and redhaired and is a girl’ (no matter whether she is nice or naughty).

‘big’ is compatible to ‘blonde’ because $1+02=12>\emptyset$, so you may be big and blonde at the same time.

‘big’ is incompatible to ‘little’ because $1+2=\emptyset$, so you cannot be big and little at the same time.

‘little redhaired’ is subordinate to ‘redhaired’ because $23+03=23$.

If the murderer is known to be blackhaired, then the little redhaired girl is beyond suspicion because $2301><01$.

Thus the ordinal fraction algebra does Sherlock Holmes’ work, which is logical reasoning. Ordinal fraction algebra is *artificial intelligence*.

8. **Databases**

Let the table in the previous example be supplemented by a list of names.

```
1111 Berta
1112 Klavs
1121 Trine
1122 Niels
1211 Inger
1212 Tomas
1221 Marie
1222 Peter
1311 Rikke
1312 Rumle
1321 Gerda
1322 Allan
2111 Nanna
2112 Frede
2121 Sofie
2122 Kevin
2211 Lotte
2212 David
2221 Birte
2222 Jakob
2311 Frida
2312 Søren
2321 Sonja
2322 Carlo
```

If you are interested in little redhaired girls, all records *compatible* to `2301` are extracted from this database.

```
0001 girl
001 nice
002 naughty
03 redhaired
2 little
2311 Frida
2321 Sonja
```

This extract contains everything of interest: The *superordinates* verifies `2301` as ‘little redhaired girl’. The *subordinates* is the complete list of little redhaired girls, (Frida, Sonja). The *coordinates* shows the difference between them, Frida being nice while Sonja is naughty.

Statements concerning elements of the database matches relations between the corresponding ordinal fractions.

$2122 \leq 0002$	‘Kevin is a boy’
$1211 \leq 001+02$	‘Inger is nice and blonde’
$2321 \leq 2301$	‘Sonja is a little redhaired girl’

9. **Access rules**

Records compatible to `0001` are written to the left and records compatible to `0002` are written to the right.

0001 girl	0002 boy
001 nice	001 nice
002 naughty	002 naughty
01 blackhairede	01 blackhairede
02 blonde	02 blonde
03 redhaired	03 redhaired
1 big	1 big
2 little	2 little
1111 Berta	1112 Klavs
1121 Trine	1122 Niels
1211 Inger	1212 Tomas
1221 Marie	1222 Peter
1311 Rikke	1312 Rumle
1321 Gerda	1322 Allan
2111 Nanna	2112 Frede
2121 Sofie	2122 Kevin
2211 Lotte	2212 David
2221 Birte	2222 Jakob
2311 Frida	2312 Søren
2321 Sonja	2322 Carlo

If you are in charge of the girls, you will only be interested in the sub-database to the left, and your collegue in charge of the boys will only be interested in the sub-database to the right. The following simple rules apply in order to protect your collegue against your interference into his business:

1.You have no access to incompatibles.
2.You have read-access to compatibles.
3.You have write-access to subordinates.

A><B **:** A has no access to B
A≤≥B **:** A has read-access to B
A ≥B **:** A has write-access to B

10. **Roman numbers**

Interpretation of ordinal fractions may be done by assigning a ‘variable-name’ to each digit-position. For example:

```
2000 matches  M=2
0300 matches  C=3
0001 matches  I=1
```

The variablenames `M, C, X,` and `I` are chosen because they are roman numerals meaning ‘thousand’, ‘hundred’, ‘ten’ and ‘one’. The ordinal fraction

```
2301
```

mathes the logical proposition

```
M=2 AND C=3 AND I=1
```

The ordinal fraction notation has the same advantage above logical algebra as arabic numbers have above Roman numbers: it’s more compact and simpler to do calculations.

Variable-names prevent a unified approach to arrays and hierarchies. However, they are so deeply rooted in computer-science and mathematics that a suggestion to abolish variable-names are strongly resisted.

It is therefore remarcable that the use of variable-names in mathematics i probably due to an error of translation!

In Euclid’s ‘Elements’ a triangle is denoted ‘alpha beta gamma’. Translated into latin it became ‘ABC’.

Algebra evolved and took over the convention to use one-letter identifiers for variables.

Programmers began to use many-letter identifiers having mnemonic significance.

So, here we are. But Euclid’s ‘alpha-beta-gamma’ are ordinal numbers, not words.

As arabic numerals were not invented, the triangle was not called ‘1 2 3’, which would be the obvious translation today.

11. **References**

Bo Jacoby: *Ultrafleksibel databasestruktur og kunstig katolicisme,* NordDATA89. (Complete recipe for making your own artificial catholic).

Bo Jacoby: *Graf, relation og ordinalbrøk*, 1988 (unpublished, on how to number partially ordered sets).

Bo Jacoby: *BHR Videnbank*, *Brugervejledning*, 1988 (unpublished, User's Guide to a big database).

Bo Jacoby: *Fjerdeproportionalen mellem ordinalbrøker*, 1988 (unpublished, on how to represent analogies).

Bo Jacoby: *Den kunstige katolik*, Computerworld nr 24, 1988 (Announcing artificial catholicism).

Bo Jacoby: *Integration og ordinalbrøk*, 1988 (unpublished, generalizing the mathematical concept of integration).

Bo Jacoby: *From cardinal numbers to ordinal fractions*, 1986 (unpublished, number theoretical approach).

Bo Jacoby: *Rundskrivelse om ordinalbrøker*, 1986 (unpublished, summary).

Bo Jacoby: *Ordinal Fractions - the algebra of data*, 1985 (rejected IFIP'86-paper, similar to the NordDATA90-paper).

Bo Jacoby: *Ordinal Fractions*, 1984 (rejected BIT-paper, a thorough theoretical treatment)

Bo Jacoby: *Århundredets matematiske nyskabelse*, Nyboder avisen nr 10, 1984 (announces the theory of ordinal fractions).

March 1990
Bo Jacoby
J.A.Schwartzgade 4
DK-2100 København Ø
tel +45 31265203

Retyped into MS-Word November 2000
Bo Jacoby
bojacoby@worldonline.dk
tel +45 35265203

Saved as Rich Text File, June 2001.