Serge GLADKOFF ; Chez Mme Luciani ; 39,rue du Moulin ; 91120 PALAISEAU ; FRANCE

NON-STANDARD ANALYSIS AND NUMERATION SYSTEMS

Abstract :

I will show how non-standard analysis can help in describing numeration systems, such as that used by fixed-point arithmetics in computers. To achieve the non-standard extension of the total order, instead of the usual definition using ultrafilters, a lexicographical ordering will be used.

Summary

0.Notations

Because of the incompletude of the printing machine I could use, to denotate the usual mathematical characters, I will use the following :

Existence quantifier	: Ɇ
Universal quantifier	: ¥
Membership relation	: ∈
Inclusion relation	: c
Intersection relation	: ∩
Union relation	: U
Empty set	: Ø
Power operation	: **
Logical "and" operator	: ^
Logical "or" operator	: v
Logical "not" operator	: ¬
Indices	: a[i]
Mapping	: f(x) f(x,y)
Indexed mapping	: f[i](x[j])

1.Construction of the non-standard extension of a set, with its properties

Definitions

E is a set of "standard objects" or "standard numbers"
X is a set of "indices"
A is the set of all the mappings X-->E
A=E**X={f:X-->E}

Identification

For A to be considered an extension of E , we must identify the elements of E with some elements of A .
The elements of A identified to those of E are called the "standard elements" of A , the others the "non-standard elements" of A .
If X=Ø , there is only 1 element in A : not enough.
If X={x} , there is a bijection between E and A : nothing new.
We will no more consider these two cases.
If X contains at least 2 elements, A contains at least as many elements as the set P(E) of all the subsets of E , that is strictly more than E .
A method to make the identification having been chosen (several are possible), we can then write EcA .

Properties

We want to transfer the properties of the "small" set E to the "big" set A . We can translate :

elements
  e[1],e[2],...e[n]∈E   f[1],f[2],...f[n]∈A
properties
  p[1](e[1])   p[1](f[1])="{x∈X:p[1](f[1](x))}∈FcA"
  p[2](e[1],e[2])   p[2](f[1],f[2])="{x∈X:p[2](f[1](x),f[2](x))}∈FcA"
  where F is a chosen fixed subset of P(X) .

What needs to be F ?

That depends of which properties we want to be transfered. Without any hypothesis on F , equality is transfered to an equivalence relation, and we take the classes : A=(E**X)/F

Examples:

Transfer of a reflexive relation ~ :
  reflexivity of ~ on E : ¥e∈E e~e
  then : ¥f∈A ¥x∈X f(x)~f(x)
  then : ¥f∈A {x∈X : f(x)~f(x)}=X
  so that the transfer just needs : X∈F

Transfer of a symmetric relation ~ :
  symmetry of ~ on E : ¥d∈E ¥e∈E d~e==>e~d
  then : ¥f∈A ¥g∈A ¥x∈X f(x)~g(x)==>g(x)~f(x)
  then : ¥f∈A ¥g∈A {x∈X : f(x)~g(x)}c{x∈X : g(x)~f(x)}
  so that the transfer just needs : ¥P∈F PcQ==>Q∈F

Transfer of an antisymmetric relation ~ :
  antisymmetry of ~ on E : ¥d∈E ¥e∈E (d~e)^(e~d)==>d=e
  then : ¥f∈A ¥g∈A ¥x∈X (f(x)~g(x))^(g(x)~f(x))==>f(x)=g(x)
  then : ¥f∈A ¥g∈A {x∈X : f(x)~g(x)}∩{x∈X : g(x)~f(x)}c{x∈X : f(x)=g(x)}
  so that the transfer just needs : ¥P∈F ¥Q∈F P≥Q∈F
    and : ¥P∈F PcQ==>Q∈F
    i.e.: F is a filter

Transfer of a transitive relation ~ :
  transitivity of ~ on E : ¥a∈E ¥e∈E ¥b∈E (a~e)^(e~b)==>a~b
  then : ¥f∈A ¥g∈A ¥h∈A ¥x∈X (f(x)~g(x))^(g(x)~h(x))==>f(x)~h(x)
  then : ¥f∈A ¥g∈A ¥h∈A {x∈X : f(x)~g(x)}∩{x∈X : g(x)~h(x)}c{x∈X : f(x)~h(x)}
  so that the transfer just needs : ¥P∈F ¥Q∈F P≥Q∈F
    and : ¥P∈F PcQ==>Q∈F
    i.e.: F is a filter

Transfer of a total relation ~ :
  totalness of ~ on E : ¥d∈E ¥e∈E (d~e)v(e~d)
  then : ¥f∈A ¥g∈A ¥x∈X (f(x)~g(x))v(g(x)~f(x))
  then : ¥f∈A ¥g∈A {x∈X : f(x)~g(x)}∪{x∈X : g(x)~f(x)}=X
  so that the transfer just needs : P∈F<==>¬(X-P∈F)
    and : ¥P∈F PcQ==>Q∈F
    i.e.: F is an ultrafilter
  thus : {x∈X : f(x)~g(x)} and {x∈X : g(x)~f(x)} contain two complementary subsets of X , one of which being in F , with the sets including it.
  remark : both complementary subsets cannot be in F , otherwise their empty intersection would also be in F , and the resulting system would be inconsistent, since the properties could be accepted even if true for no x∈X .

General case :

If F is an ultrafilter, all properties can be transfered, and the theorems on A can be demonstrated as on E , using the classical logic ; in particular, the identification can be done by the equivalence class of the constant applications :
e[0]"="{f∈E**X:{x∈X:f(x)=e[0]}∈F}

If F is only a filter, the properties expressed with an irreductible "or" or "there exists" or a "not" which is not in terminal position are not transfered ; it seems to be linked with the non-transfer of the excluded-third-case principle, so that the intuitionnistic logic would work, but not the classical.

The transfer of preorder, equivalence, and order relations need only a filter. But a total order would be transfered to a partial order, unless the filter is an ultrafilter. But this partial order can be completed into a total order using other methods, which will be studied further on.

2.Application to numeration systems : filter or ultrafilter ?

Definitions

E : set of digits, finite, totally ordered
X : set of index
If b=Card(E) , we can see a base-b number as a mapping X-->E

The order on the numbers should be deduced from the order on the digits, and should be total.
Let's take X finite, since the number of places where you can write a digit is always finite in practice, though it can be big.
Then, with E**X , you can represent (Card(E))**(Card(X)) different numbers, that can still be chosen at your convenience, several conventions being used in practice.

But what happens with an ultrafilter ?

On a finite X , all the ultrafilters are principal (i.e. contain exactly one singleton).
If you take the classes (E**X)/F , all the mappings that have the same value at x[F] (where {x[F]}∈F) are equivalent, so that we have in fact as many numbers as we have digits : no new numbers.

Could we take an infinite X ?

Example :

with X=N , there are infinite numbers and no infinitesimals
with X=Z , there are infinite numbers and infinitesimals

But there are some drawbacks :

1/ This means some circularity : to construct N , we need N .
2/ Assume I want to decide if f≤g :
   -if the ultrafilter F is principal, with {x[F]}∈F :
      I just need to look if f(x[F])≤g(x[F])
   -if the ultrafilter F is not principal, there are 3 cases :
      with K={x∈X : f(x)≤g(x)}
      1-if K is finite, it is a finite union of singletons, which are not in F , so that K is not in F .
      2-if X-K is finite, K is a finite intersection of complements of singletons, which are in F, since the singletons are not, so that K is in F .
      3-neither K nor X-K is finite, and I must decide which of them is in F . That means that before I can compare any f,g I must have done an infinite (non-denumerable) choice between the parts of X and their complement. This is impossible in practice for anybody and any computing machine.
3/ We must allow F to be a filter, and will then be allowed to make only a denumerable choice (which can be defined by a certain algorithm) to decide if a part is in F or its complement is in F , knowing that both cannot be in F , but it is possible that neither is in F (it is even almost always the case).
   But the order is not total, because of the undecided pairs.

Remark

Assume X is finite and F is a filter which is NOT an ultrafilter.
We find the same type of discussion in 3 cases than for an ultrafilter on an infinite X .
If this could be more precisely formalized, it could perhaps be used to simulate or imitate proofs involving the non-denumerable choice with a system that is finite, so that every calculation and case-checking would be assured to terminate in finite time.

E=<0,1,2,3>
X=<I,II>

F=<X>
(∀y∈E)(∀x∈X) f,(x)=y

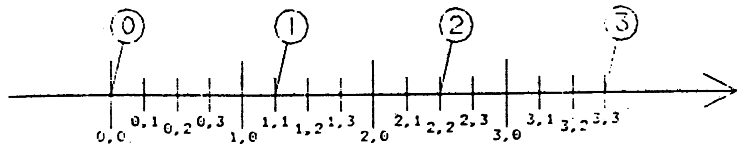

FIGURE 1

E=<0,1,2,3>
X=<I,II>

F=<<I>,X>
(∀y∈E) f,(I)=y f,(II)=0

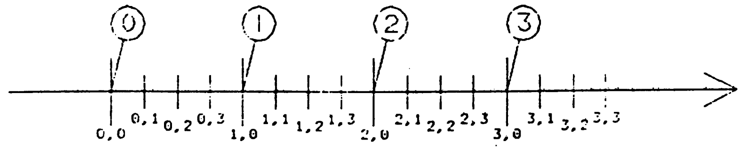

FIGURE 2.0

E=<0,1,2,3>
X=<I,II>

F=<<I>,X>
(∀y∈E) fy(I)=y fy(II)=1

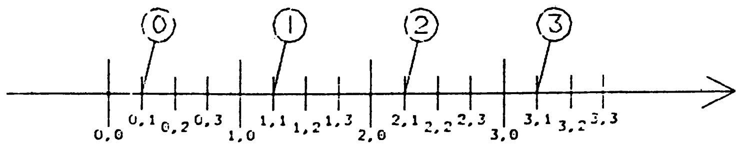

FIGURE 2.1

E=<0,1,2,3>
X=<I,II>

F=<<I>,X>
(∀y∈E) f,(I)=y fy(II)=2

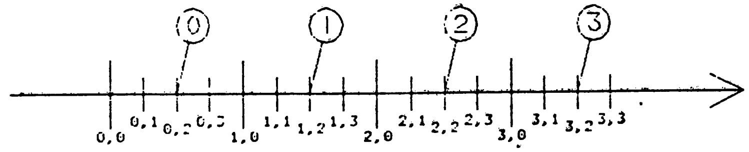

FIGURE 2.2

E=<0,1,2,3>
X=<I,II>

F=<<I>,X>
(∀y∈E) f,(I)=y f,(II)=3

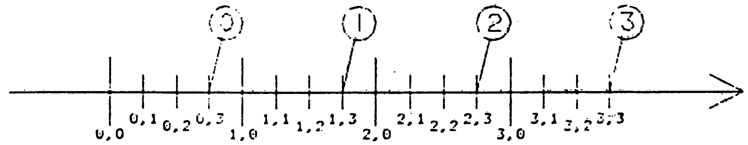

FIGURE 2.3

E=<0,1,2,3>
X=<I,II>

F=<<II>,X>
(∀y∈E) f,(I)=0 f,(II)=y

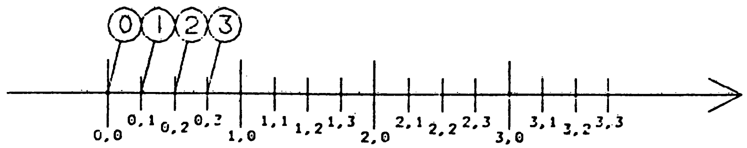

FIGURE 3.0

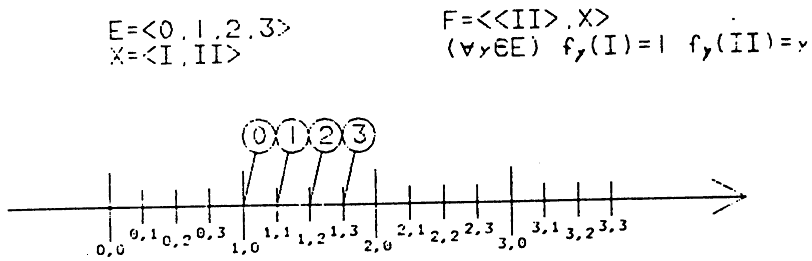

FIGURE 3.1

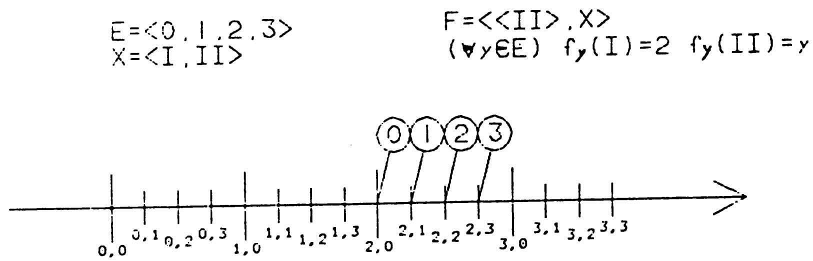

FIGURE 3.2

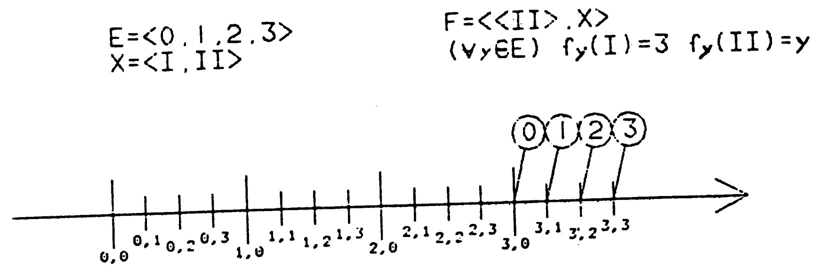

FIGURE 3.3

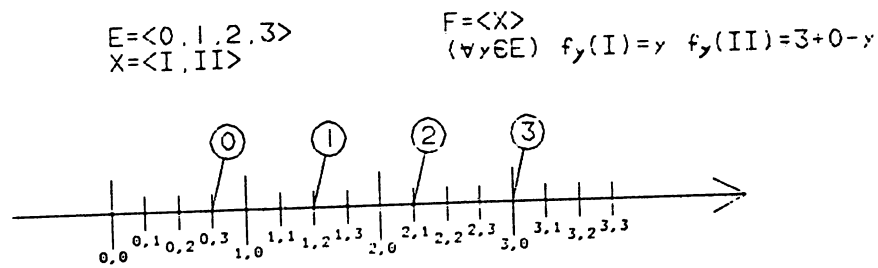

FIGURE 4

3. Application to computers : use of a lexicographical order

E={0,1} is the set of logical values of the elementary bits
X is the set of indexes of the bits in a machine word
(usually card(X) will be a power of 2 , often 8,16,32,64)

We now take E**X , and F={X} , which is a (degenerate) filter, but not an ultrafilter if X has more than one element. Thus, we have the full richness of new numbers in A=(E**X)/F=E**X , and the order relation ≤ can be transfered, but it is not total.

But this partial order can be enriched, so as to become total : take a total order on X (finite), and on A the lexicographical order induced by the order on the indexes : it is compatible and richer than the order generated by the filter, and it is total. This was possible because X was finite (in fact, we needed only that X has a maximum to define a lexicographic order).

There are many ways to define the identification function between the "new numbers" and the "standard" ones, as can be seen on the figures 1 to 4 .

Other possible applications :

-Integer Double- or Multi- Precision :
E is the set of single precision integers (standard numbers), X has 2 elements for double precision, or "n" for multi-precision, A=E**X is the set of double or multi-precision numbers, with an identification function similar to Fig. 3.0 (if positive unsigned integers) or to Fig. 3.2 with E={-2,-1,0,1} (if signed integers)

-Fixed-point real numbers
similar to Fig. 2.2

-Floating-point real numbers
similar to Fig. 2.2 but with an non-constant "density", the new numbers being more numerous near 0 , and the big numbers being more and more far from each other, in an approximately exponential manner.

4. Conclusion

There are many ways to use such non-standard analysis, I tried to show that the often neglected finite models that can be built are usable in a great variety of situations, in particular to get adequate models of the calculations made in computers. Other approaches can be found in the bibliography given.

5. Bibliography

L.Haddad : "Condorcet et les Ultrafiltres" (Colloque de Luminy-1985)

A.Marquetty : "Approche algébrique de l'analyse non-standard" (R.A.E Vol.8-n°1-1982/83-p.39-57)

A.Marquetty,S.Gladkoff : "L'analyse non-standard expliquée aux mathématiciens non logiciens" (to appear in "Eleutheria" : Pr.S.P.Zervos - Athens - Greece)

J.Mycielski : "Finistic real analysis" (R.A.E Vol.6-1980/81)

J. Harthong : "Elements pour une théorie du Continu" (Astérisque 109-110 (1983) p 235-244)

E Reeb : "La mathématique non-standard, vieille de 60 ans ?" (Publication IRMA 1978)