Supplementary Notes for MAC-lectures September 3–8

September 5, 2003

Type Tables

From the *classical* mathematics we know some *classical* values, e.g.

The positive integers: 1, 2, 3, etc.
In the Mac System $\mathbf{Z}^+$ denotes the set containing all 'classical maps weakly/strongly representing' a positive integer value (Z stands for the german "zahlen"). Before we reach chapters 9 and 10 we cannot understand the phrase 'classical maps weakly/strongly representing', but it helps to know that

- The Map program actually *represents* all the mathematical expressions in our mathematical system (e.g. the Mac System) by means of so-called *maps*. Hence, Map will (and we may) interpret the expressions 1, 2, 1+1, 2–0, 3 etc. as being 'classical maps'!
- The Map program may have several (weak or strong) representations (maps) for expressions with the *same* value, but all these maps represent that value. Hence, Map may represent the number 2 in different ways, as well as the term 1+1, but these maps all represent the value 2.

If you now are ready to accept that e.g. 1+1 is a 'classical map representing' a positive integer, I can tell you that this fact is expressed by [$1+1 \in \mathbf{Z}^+$] in the Mac System.
Hence, the set $\mathbf{Z}^+$ contains all terms being **equal** to one of the integers 1, 2, 3, 4 etc.

The natural numbers: (Terms equal to) 0, 1, 2, 3, etc. belong to the set **N**.

The negative integers: (Terms equal to) -1, -2, -3, etc. belong to the set $\mathbf{Z}^-$

The integers: (Terms equal to) …, -3, -2, -1, 0, 1, 2,… belong to the set **Z**

The decimal fractions: (Terms equal to e.g. the numbers above, but also some more – c.f. next page) belong to the set **D**

The boolean values: (Terms equal to) T, F belong to the set **B**.
Furthermore, [$T \in \mathbf{T}$] and [$F \in \mathbf{F}$] hold.

The exception: (Terms equal to) • belong to the set **X**

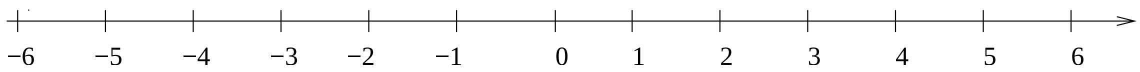
Note that • is considered a classical value/map, contrary to the value ⊥ which is **non-classical** and does not belong to *any* set! If we e.g. have to find the values of the terms

$[\ T + 5\ ]^{\cdot}$ and $[\ \perp + 5\ ]^{\cdot}$

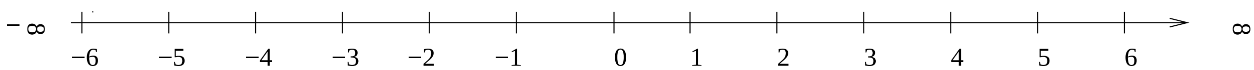
we may look at page 607 and use the type table, row 2 coloumn 3, for the first expression (saying that an element in **B** plus an element in **D** gives an element in **X**, i.e. •). Since ⊥ does not belong to any set, we cannot, however, use the type table for the second expression, but the rule below the table is applicable.

Real Numbers and Decimal Fractions

The set of real numbers is often illustrated by a number axis, on which only the integers are marked:



If you have an infinite sequence of increasing/decreasing real numbers of the form 1.0, 1.5, 1.75, 1.875,.. where the difference from 2 is halved in each step, it is not difficult to see that the sequence *converges to* (i.e. is from a certain step arbitrary close to) a number (2), but how about the sequences 1, 2, 3, 4,.... or -1, -2, -3, -4,... ?
The latter sequences do *not* converge to any real number, so therefore the set of real numbers is often supplemented by two *non-real* numbers: ∞ ('infinity') and $-\infty$ ('minus infinity'). These two numbers are respectively larger than and smaller than any real number, and now 1, 2, 3, 4,... converges to a number (∞) and so does -1, -2, -3, -4,... ($-\infty$). This extended number set may be illustrated this way:



Besides the integers, what numbers are contained in the set of real numbers?

Decimal Fractions: Apart from digits and maybe a sign symbol, we use **a decimal point** to obtain numbers like -0.0012045, 120.45, +1204500.000 (the latter being equal to the integer 1204500).

Rational Numbers: Denoting numbers as **ratios between integers**, we now have the decimal fractions -12045/10000000, 12045/100, +1204500/1 as well as some more numbers, e.g. 1/3 =0.3333333333333......

Irrational Numbers: Although we may come arbitrarily *close* to any real number by using rational numbers, there are *some* real numbers that are not rational. As an example π = 3.14159265358979323846.... is irrational, although the rational numbers 22/7, 2199/700, 219911/70000,... come closer and closer to π.

Klaus might have chosen to include all these numbers (they can be represented by 'maps') in the Mac System, but since this course is not 'only' about mathematics, but also computation, he chose the (**exact**/classical) decimal fractions and some **floating**/hardware-represented decimal fractions as the most general type of numbers in the Mac System, and **included** exact and floating versions of ∞ and $-\infty$ in the set **D** of decimal fractions!
Assume that you have a computer that computes with digits (we don't want to look at bits), and the registers may contain 3 decimal digits plus information about the sign (+,–) and the position of the decimal point. That computer cannot represent a number like -0.0012045, but the digits in 120, the -, and the position of the decimal point can be stored:

Mantisse: m = -1.20 (normalised such that $1 \le | m | < 10$), **Exponent**: e = -3

Reading this 'register' we see the **floating** decimal fraction corresponding to $\mathbf{m \cdot 10^{e}}$, i.e. the Mac number

-0̸.0̸0̸120F

When the 0's are *not* stored, they are denoted 0̸. Hence, the number -0̸.0̸0̸120F reveals that only 3 digits have been stored (the **precision** of the number is 3), and that it is a **floating(due to 'F')**/stored representation of the **exact**/classical decimal fraction -0.00120 (which has **infinite precision**, since it is the same as -0.0012000000000...).

Decimal Fractions and Typetables

The **exact** decimal fractions have no 'F' or stroked 0's, and they are found in the set $\mathbf{D}^{\cdot}_{\infty}$, since the subscript on **D**-sets denotes the precision. $\mathbf{D}^{\cdot}_{\infty}$ consists of the classical decimal fractions and thus $\pm\infty$ are **not** in this set!
The **exact** numbers ∞ and $-\infty$ have infinite precision, too, since they are not stored in a register, hence they are found in the sets $\mathbf{D}^{\infty}_{\infty}$ and $\mathbf{D}^{-\infty}_{\infty}$, respectively.
The exact decimal fractions and the exact ∞ and $-\infty$ may be **stored** on a computer with space for m digits or on a computer with space for (a different number) n digits. The subscripts on the **D**-sets to which these **floating** decimal fractions belong are then changed from ∞ to m or n.

Hence,

[-0.00120 $\in \mathbf{D}^{\cdot}_{\infty}$], [-$\emptyset.\emptyset\emptyset$120F $\in \mathbf{D}^{\cdot}_{3}$], [-$\emptyset.\emptyset$0120F $\in \mathbf{D}^{\cdot}_{4}$],
[-$\infty \in \mathbf{D}^{-\infty}_{\infty}$], [-$\infty$@3 $\in \mathbf{D}^{-\infty}_{3}$] and [-$\infty$@4 $\in \mathbf{D}^{-\infty}_{4}$]

since '@' is the operator, which one may use to round a number (e.g. -∞) to a floating number of a certain precision (e.g. 3 or 4), i.e. -0.00120@3 $\equiv$ -$\emptyset.\emptyset\emptyset$120F.

We may now use the type tables in volume 3 for checking whether, e.g. the sum of decimal numbers with different *finite* precisons becomes a number:

-$\emptyset.\emptyset\emptyset$120F + -$\emptyset.\emptyset$0120F $\equiv \bullet$ (Type Table, p.607, 8'th row 9'th coloumn)

or whether an exact or floating version of ∞ added to a decimal fraction gives ∞ (it does not!).