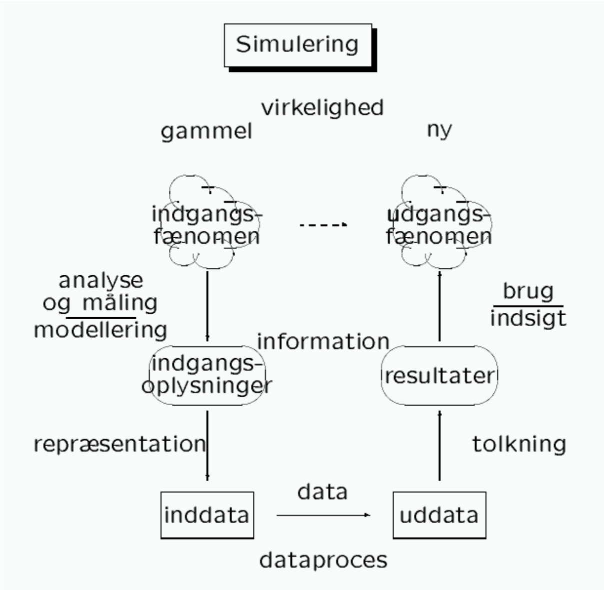
Supplementary Note for the MAC-lecture November 3

November 10, 2003

Strong and Weak Representation

What do you think of this MAC Course? Well, probably there are positive as well as (hopefully only few) negative aspects of this course, so maybe you would like the computer to help you evaluate the course?
This **is** possible! But as shown on the course Datalogi 0 (2003/04), you have to **analyze and measure** your feelings in order to specify some information ('Indgangsoplysninger') that can be **represented** by the computer as input ('inddata') for the computing proces:



Hence, the feeling that

> The MAC Course has *better* teachers and lecture notes than the other DIKU courses, but it requires *more* work to understand the teaching material,

has to be analysed in order to specify representable information, as e.g.

Teachers and Lecture Notes:	1206.5%	better
Work Required:	1004.3%	more

If we consider these two aspects equally important, we may now specify information on the computing proces, using three variables:

Teaching = 12.065; Work = 10.043; Result = (Teaching - Work)

Being **an electronic device** the computer represents all information by means of electronic charges and voltages(!), but in order to **describe** machine representation to non-technicians the concept **bits** was introduced many years ago. A bit is a binary digit, usually describing *either* low voltage (bit = 0) *or* high voltage (bit = 1), where the different electronic units may differ in their interpretation of what is low/high voltage, e.g. to handle noise on the connections properly (cf. page 321).

Unfortunately, describing the representation in terms of bits may also become rather awkward. As an example we may consider the exact decimal fraction 10.043, where the representation on most computers (satisfying the IEEE Standard 754) can be described by the following 32 bits (unless we have specified that the computer should use more bits, e.g. 64 for double precision):

> $\boxed{\text{0 10000010 01000001011000000100001}}$
> The first bit describes the sign (= +), the next eight bits the exponent
> (= 3 in Excess-127 notation) and the rest describes a major **fraction** of the mantissa
> (= **1.**01000001011000000100001 in the Base-2 system). Hence, 10.043 is 'represented' as:
> $+ \ (1 + 2^{-2} + 2^{-8} + 2^{-10} + 2^{-11} + 2^{-18} + 2^{-23}) \cdot 2^3 \quad (= 10.04300022... \ )$

Unfortunately 10.043 cannot be represented exactly by a **finite** number of bits, and we must accept a minor **rounding error**, which *may* have a major impact on the results of large technical calculations. Since this, however, is the topic of 'Scientific Computing', we will in this note represent our input numbers on a decimal machine with a 3-digit mantissa:

> $[\text{S ' } 10.043 \equiv 10.0\text{F}]^{\cdot}$ (i.e. $\boxed{+ \ 1 \ 1.00}$, interpreted as: $[+ \ 1.00 \cdot 10^1 \equiv 10.0 \equiv \text{W ' } 10.0\text{F}]^{\cdot}$)
> $[\text{S ' } 12.065 \equiv 12.1\text{F}]^{\cdot}$ (i.e. $\boxed{+ \ 1 \ 1.21}$, interpreted as: $[+ \ 1.21 \cdot 10^1 \equiv 12.1 \equiv \text{W ' } 12.1\text{F}]^{\cdot}$)

The function S shows how 'the **strong (standard)** representation' is on this version of Map (requiring the mantisse **m** to satisfy $1 \leq \mathbf{m} < 10$). However, computing the value of the variable Result (i.e. 12.1F - 10.0F) the computer must be able to interpret a representation, which is **not** a strong representation(!), since the mantissae are just subtracted when the exponents are equal:

> $[12.1\text{F} - 10.0\text{F} \equiv 02.1\text{F}]^{\cdot}$ (i.e. $\boxed{+ \ 1 \ 0.21}$, with the same W-interpretation as 2.10F, i.e. $\boxed{+ \ 0 \ 2.10}$)

Hence, the 'the interpretation function' W has to accept **strong as well as weak** representations of numbers, but Map may **normalize** the intermediate results by a so-called **retract function R**:

> $[\text{S ' } 2.1 \equiv 2.10\text{F}]^{\cdot} \quad [\text{W ' } 2.10\text{F} \equiv 2.1]^{\cdot} \quad [\text{W ' } 02.1\text{F} \equiv 2.1]^{\cdot} \ [\text{R ' } 0.21\text{F} \equiv \text{S ' } (\text{W ' } 0.21\text{F}) \equiv 2.10\text{F}]^{\cdot}$

Using the function W, Map interprets and prints the result of our computing process as 2.1 (cf. the figure from Datalogi 0) and we may use our insight to conclude that MAC is more than double as good as other DIKU courses!

Representation by Maps

Whereas bits or digits may be adequate for describing numbers, we need something more to describe **all** possible mathematical terms. However, as Klaus Grue has shown by his Map Theory:

> **All** mathematics can be described by means of
> **1)** The two ternary operators 'substitution' and 'case' (defined by the axioms page 326),
> **2)** the two binary operators 'λ' and 'apply',
> **3)** the nullary operators $[\perp]^{\cdot}$, $[\text{ N }]^{\cdot}$ ('not a function *nor* $\perp$ *nor* variable'),
> $[\text{ x }]^{\cdot}$, $[\text{ y }]^{\cdot}$,... (representing *free* variables), and
> **4)** the Existence Quantifier $[\ \bar{\exists}\ ]^{\cdot}$ and the Choice Operator $[\ \bar{\varepsilon}\ ]^{\cdot}$ (introduced in Chapter 18)

But how can e.g. a natural number like 4 be described without having bits or digits???
Well, e.g. by lambda-functions requiring 4 applications in order to produce something else than a lambda-function:

> $[\text{ S ' } 4 \equiv \lambda\text{x. } \lambda\text{x. } \lambda\text{x. } \lambda\text{x. N}]^{\cdot}$
> $[\text{ W ' } (\lambda\text{x. } \lambda\text{x. } \lambda\text{x. } \lambda\text{x. N}) \equiv \text{W ' } (\lambda\text{x. } \lambda\text{x. } \lambda\text{x. } \lambda\text{x. } \perp) \equiv 4]^{\cdot}$

But how can e.g. the successor-operator be described, when the natural numbers are represented this way???
Well, use the λ- operator, the substitution operator and a free variable:

$[\ \mathrm{S}\ \text{'}\ (4^+) \equiv \langle\ \lambda \mathrm{x}.\ \mathrm{y}\ \ |\ \ \mathrm{y} := \mathrm{S}\ \text{'}\ 4\ \rangle \equiv \lambda \mathrm{x}.\ \lambda \mathrm{x}.\ \lambda \mathrm{x}.\ \lambda \mathrm{x}.\ \lambda \mathrm{x}.\ \mathrm{N}]^{\cdot}$

As seen in next week's exercises, this representation of natural numbers may be extended to integers by means of the λ- and the case-operators:

$[\ \mathrm{S}\ \text{'}\ (\text{-}2) \equiv \lambda \mathrm{z}.\ \mathrm{case}(\mathrm{z}, (\mathrm{S}\ \text{'}\ 0), (\mathrm{S}\ \text{'}\ 2))]^{\cdot}$ (i.e. 0 - 2 is represented by means of S ' 0 and S ' 2)
$[\ \mathrm{W}\ \text{'}\ (\lambda \mathrm{z}.\ \mathrm{case}(\mathrm{z}, (\mathrm{S}\ \text{'}\ 0), (\mathrm{S}\ \text{'}\ 2))) \equiv \mathrm{W}\ \text{'}\ (\lambda \mathrm{z}.\ \mathrm{case}(\mathrm{z}, (\mathrm{S}\ \text{'}\ 3), (\mathrm{S}\ \text{'}\ 5))) \equiv \text{-}2\ ]^{\cdot}$

and addition, subtraction, multiplication and all other operations on integers may be described as well!! Since the **values** of **all** mathematical terms are represented by **a function** (e.g. S ' (-2)), **N** (e.g. S ' 0) *or* $\perp$ these nullary operators have been given the name **maps** by Klaus Grue.

Map Representations in Chapter 9

Naturally, we do not find **map**-representation of **all** mathematical terms in Chapter 9, but the possible **values** of terms are described in terms of maps(!), and a representation of the ternary **if**-operator is suggested.
In order to **simplify the notation** three operators are defined in terms of the λ-, case-, and apply-operators:

$[\ \mathrm{x} \therefore \mathrm{y} \doteq \lambda \mathrm{z}.\ \mathrm{case}(\mathrm{z},\mathrm{x},\mathrm{y})\ ]^{\cdot}$
$[\ \mathrm{x}\ \mathrm{Head} \doteq \mathrm{x}\ \text{'}\ \mathrm{N}\ ]^{\cdot}$
$[\ \mathrm{x}\ \mathrm{Tail} \doteq \mathrm{x}\ \text{'}\ (\lambda \mathrm{z}.\ \mathrm{N})\ ]^{\cdot}$

Choosing

$[\ \mathrm{S}\ \text{'}\ \mathrm{T} \equiv \mathrm{N}\ ]^{\cdot}$ (and $[\ \mathrm{W}\ \text{'}\ \mathrm{x} \equiv \mathrm{T}\ ]^{\cdot}$ only if $[\ \mathrm{x} \equiv \mathrm{N}\ ]^{\cdot}$)
$[\ \mathrm{S}\ \text{'}\ \mathrm{F} \equiv \mathrm{N} \therefore \mathrm{N}\ ]^{\cdot}$ (and $[\ \mathrm{W}\ \text{'}\ \mathrm{x} \equiv \mathrm{F}\ ]^{\cdot}$ only if x is a function with $[\ \mathrm{x}\ \mathrm{Head} \equiv \mathrm{N}\ ]^{\cdot}$)
$[\ \mathrm{S}\ \text{'}\ \bullet \equiv (\ \mathrm{N} \therefore \mathrm{N}\ ) \therefore (\ \mathrm{N} \therefore \mathrm{N}\ )\ ]^{\cdot}$ (and $[\ \mathrm{W}\ \text{'}\ \mathrm{x} \equiv \bullet\ ]^{\cdot}$ only if $[\ \bar{\mathrm{X}}\ \text{'}\ \mathrm{x} \equiv \mathrm{T}\ ]^{\cdot}$)
$[\ \mathrm{S}\ \text{'}\ (\mathrm{if}(\mathrm{x},\mathrm{y},\mathrm{z})) \equiv \mathrm{case}(\mathrm{x},\mathrm{y},\mathrm{case}(\mathrm{x}\ \mathrm{Head},\mathrm{z},\mathrm{S}\ \text{'}\ \bullet))\ ]^{\cdot}$

one may e.g. show that the **map**-representation of the if-operator acts as we want on the strong representations of T, F and $\bullet$.
The above function $[\ \bar{\mathrm{X}}\ ]^{\cdot}$ is one of several **type functions** (Section 9.13) used by the function W to distinguish between representations of elements in the sets **B**, **X**, **D**, **E** and **pairs**.
The **map**-representation of **pairs** and the element $[\ \langle\ \rangle\ ]^{\cdot}$ in **E** is described in Section 9.11, whereas the **map**-representation of elements in **D** is described in section 9.10.

We may now compute the terms $[\ \mathrm{T}\ \text{'}\ \perp\ ]^{\cdot}$ and $[\ (\ 1 :: 5\ )\ \text{'}\ 6\ ]^{\cdot}$ from the previous supplementary note **by using map-representations**! The first term is simple due to the axiom $[\ \mathrm{N}\ \text{'}\ \mathcal{B} \equiv \mathrm{N}\ ]^{\cdot}$ (page 326), and the result is interpreted by the Map-computer as T ($\equiv$ W ' N).

It may be somewhat difficult to remember the **map**-representations of the symbols $\underline{0}$, $\underline{1}$, $\underline{2}$, ..., $\underline{9}$ used in the representation of decimal fractions, but try to draw the syntax trees and note how the tree corresponding to $\underline{1}$ moves **up the left subtree and down the right subtree** as we consider the representation of $\underline{2}$, $\underline{3}$, ..., $\underline{9}$.