Operations on Naturals

Prof. Daniele Gorla

Arithmetics in base 2

All operations are done like in base 10, but *modulo 2*

Es.: $(1 + 1)_2 = 10_2$

Hence, also carries and borrows work *modulo 2!!*

SUM:

In base 2, we have:

0+0 = 0, carry = 0
0+1 = 1+0 = 1, carry = 0
1+1 = 0, carry = 1

Sum (example)

Sum in base 2 the numbers 110001 and 10111.

Solution:

```
 1 1 0 0 0 1  +
   1 0 1 1 1  =
--------------
1 0 0 1 0 0 0
```

Indeed: $110001_2 = (2^5+2^4+1)_{10} = (32+16+1)_{10} = 49_{10}$

$10111_2 = (2^4+2^2+2+1)_{10} = (16+4+2+1)_{10} = 23_{10}$

$1001000_2 = (2^6+2^3)_{10} = (64+8)_{10} = 72_{10}$

Overflow

In the previous example, if the numbers were represented with 6 bits, the result would not have been representable (it asks for 7 bits) → *overflow*

No overflow if working with 7 or more bits

When summing naturals, once we fix the size of the format, we have an overflow if and only if the carry resulting from the sum of the MSBs is 1

Subtraction

Within naturals, subtraction is defined only if the minuend is at least as big as the subtrahend, that is

$m - s$ is defined only if $m \geq s$

Subtraction	Difference	Borrow
0-0	0	0
1-1	0	0
1-0	1	0
0-1	1	1

If there is a borrow and the previous bit is 1, the latter is turned into 0;

If there is a borrow and the previous bit is 0, the latter is turned into 1 and so on for all the following 0's, until we find a 1. This latter bit is turned into 0 and subtraction goes on over the modified minuend;

If no 1 is met, then the subtrahend is bigger than the minuend and so the subtraction is not possible in the naturals.

Subtraction

Examples:

```
 0            011           01011
11010 -      11000 -       101000 -
00100 =      10001 =       011001 =
-----        -----         ------
10110        00111         001111
```

$(26-4=22)_{10}$ $(24-17=7)_{10}$ $(40-25=15)_{10}$

Multiplication

As we are used to, but in base 2:

- Partial products
- Shifting of the partial products
- Sum of the shifted partial products

```
    1011 ×          11₁₀ ×
    1101 =          13₁₀ =
    ----          --------
    1011             33
   0000 -           11 -
  1011 - -        --------
 1011- - -          143₁₀
---------
10001111
```

OBS.: the result has a double length!

Division

More complex than multiplication

Same way as we are used to, but always in base 2!

147:11 = 13 reminder 4

```
147 | 11         10010011 | 1011
11  | 13          1011    |
 37               001110  | 1101
 33                 1011
  4                 001111
                      1011
                       100
```