

Laboratorio 1: I numeri naturali in OCaml

Fausto Spoto

24 gennaio 2005

I numeri naturali in OCaml

Ocaml ha i numeri naturali

Sono scritti come ci si aspetta (0, 1, 2...) e le normali operazioni aritmetiche sono definite su di essi.

Ma non sapremmo ridefinirli con pattern matching!

Provare per credere...

Ma possiamo codificarli tramite termini...

```
type natural = Z | S of natural;;
```

Esempi:

- 0 lo scriviamo Z
- 1 lo scriviamo S Z
- 2 lo scriviamo S (S Z)
- ...

sup n m se e solo se $n > m$

Pattern matching su due argomenti!

```
let rec sup n m = match (n,m) with
  (Z,b) -> false
| (S a,Z) -> true
| (S a,S b) -> sup a b;;
```

```
val sup : natural -> natural -> bool = <fun>
# sup Z Z;;
- : bool = false
# sup (S Z) Z;;
- : bool = true
#
```

add m n calcola $m + n$

Prima versione

```
let rec add m n = match (m,n) with
  (Z,b) -> b
  | (S a,b) -> S (add a b);;
```

Seconda versione

```
let rec add m n = match (m,n) with
  (Z,b) -> b
  | (a,Z) -> a
  | (S a,S b) -> S (S (add a b));;
```

```
val add : natural -> natural -> natural = <fun>
# add (S Z) (S (S (S Z)));;
- : natural = S (S (S (S Z)))
#
```

Sottrazione e massimo comun divisore

```
sub m n calcola  $m - n$ 
```

```
exception Arithmetic_exception;;
```

```
let rec sub m n = match (m,n) with  
  (a,Z) -> a  
  | (Z,a) -> raise Arithmetic_exception  
  | (S a,S b) -> sub a b;;
```

```
gcd m n calcola il m.c.d. fra  $m$  ed  $n$ 
```

```
let rec gcd m n =  
  if m = n then m  
  else if sup m n then gcd (sub m n) n  
  else gcd m (sub n m);;
```

In laboratorio!

Definite in Ocaml le seguenti funzioni (senza usare =)

<code>sup m n</code>	che calcola	$m > n$
<code>inf m n</code>	che calcola	$m < n$
<code>diff m n</code>	che calcola	$m \neq n$
<code>equal m n</code>	che calcola	$m = n$
<code>add m n</code>	che calcola	$m + n$
<code>sub m n</code>	che calcola	$m - n$
<code>mul m n</code>	che calcola	$m * n$
<code>exp m n</code>	che calcola	m^n
<code>max m n</code>	che calcola	$\max(m, n)$
<code>div m n</code>	che calcola	$\lfloor m/n \rfloor$
<code>rem m n</code>	che calcola	$m \% n$
<code>gcd m n</code>	che calcola	il m.c.d. fra m ed n
<code>fact m</code>	che calcola	$m!$
<code>prime m</code>	che calcola	" m è un numero primo"