$$f(x,y) = \begin{cases} y+1 & \text{se} \quad x=0 \\ 2+f(x-1,y) & \text{se} \quad x>0 \end{cases}$$

$$f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

$$\left(\forall n, m \in \mathbb{N} \quad f(n,m) = 2n + m + 1 \right)$$

$$(\mathbb{N}\times\mathbb{N}, \prec)$$

$$\Big(\forall n, m, n', m' \in \mathbb{N}.\ (n,m) \prec (n',m') \equiv n' = n+1 \wedge m' = m\Big)$$

$$\left\{\begin{array}{c}\vdots\\ (n,m)\\ |\\ (n-1,m)\\ |\\ \vdots\\ (0,m)\end{array}\right.$$

infiniti minimali

$$f(x,y) = \begin{cases} 2x+1 & \text{se } y=0 \\ 1+f(x,y-1) & \text{se } y>0 \end{cases}$$

$$f : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \qquad (x,y-1) \prec (x,y)$$

$$\left(\forall n,m \in \mathbb{N}.\ f(n,m) = 2n+m+1 \right)$$

Caso base

$$f(0,m) = 2 \cdot 0 + m + 1$$

$$f(0,m) = \{ \text{def } f,$$

(n,m)

$(n-1,m)$

$\vdots$

$(0,m)$

$$f(x,y) = \begin{cases} 2x+1 & \text{se } y=0 \\ 1+f(x,y-1) & \text{se } y>0 \end{cases}$$

$$\begin{array}{c|l} (x,y) & (\forall n,m,n',m' \in \mathbb{N}. \\ (x,y-1) & (n,m) \prec (n',m') \equiv \\ \vdots & n'=n \wedge m'=m+1) \\ (x,0) & (n'=n \wedge m=m'-1) \end{array}$$

$(\mathbb{N} \times \mathbb{N}, \prec)$

<u>Caso base</u>

$$f(n, 0) = 2n + 0 + 1$$

$$f(n, 0)$$

$$= \{\text{def } f,\ 1^\circ \text{ caso}\}$$

$$2n + 1$$

$$= \{\text{calcolo}\}$$

$$2n + 0 + 1$$

Caso induttivo

$\forall n, m \in \mathbb{N}$

$$f(n,m) = 2n + m + 1$$

$\Longrightarrow$ ip. induttiva

$$f(n,m+1) = 2n + (m+1) + 1$$

$$f(n,m-1) = 2n + (m-1) + 1 \quad \Big| \; m > 0$$

$$\Longrightarrow f(n,m) = 2n + m + 1$$

$$f(n, m+1)$$

$$= \{\text{def } f,\ 2^\circ \text{ caso}\}$$

$$1 + f(n, m)$$

$$= \{\text{ip. ind.}\}$$

$$1 + 2n + m + 1$$

$$= \{\text{calcolo}\}$$

$$2n + (m+1) + 1$$

$$f(n,m) = \begin{cases} m+1 & \text{se } n=0 \\ 2n+1 & \text{se } m=0 \\ 3+f(n-1,m-1) & \text{se } n>0 \\ & \text{e } m>0 \end{cases}$$

$m \geq n$

$(\mathbb{N}\times\mathbb{N}, \prec)$

se $n \geq 0$

ben fondato?

$$\begin{array}{ccc} (n,m) & \qquad & (n,m) \\ | & & | \\ (n-1,m-1) & & (n-1,m-1) \\ \vdots & & \vdots \\ (0,m-n) & & (n-m,0) \end{array}$$

$$f(n,m) = \begin{cases} \quad \ldots & \ldots \\ \quad \ldots & \ldots \\ \ldots f(n-1, m-1) \ldots & \ldots \end{cases}$$

precedenza indotta dalla def.

$$\left(\forall\, n, m, n', m' \in \mathbb{N}.\ (n,m) < (n',m') \equiv n' = n+1 \wedge m' = m+1 \right)$$

…

$$(\forall n, m.\ f(n,m) = 2n + m + 1)$$

Casi base

$$f(0,m) = 2 \cdot 0 + m + 1$$

$$f(n,0) = 2n + 0 + 1$$

1°) $f(0,m) = 2 \cdot \emptyset + m + 1$

$f(0,m)$

$= \{$def f. 1° caso$\}$

$m+1$

$= \{$calcolo$\}$

$2 \cdot \emptyset + m + 1$

2° caso

$$f(m, \emptyset) = 2m + \emptyset + 1$$

$$f(m, 0)$$

$$= \{\text{def } f,\ 2° \text{ caso}\}$$

$$2m + 1$$

$$= \{\text{calcolo}\}$$

$$2m + \emptyset + 1$$

$$f(n,m) = 2n + m + 1 \implies$$

$$f(n+1, m+1) = 2(n+1) + (m+1) + 1$$

$$f(n+1, m+1)$$

$$= \{\text{def } f,\ 3^\circ \text{ caso}\}$$

$$3 + f(n,m)$$

$$= \{\text{ip. induttiva}\}$$

$$3 + 2n + m + 1$$

$$= \{\text{calcolo}\}$$

$$2(n+1) + (m+1) + 1$$

$$f(n,m) = 2n + m + 1 \Rightarrow f(n+1, m+1) = 2(n+1) + (m+1) + 1$$

$f(n+1, m+1) = 2(n+1) + (m+1) + 1$

= {def f, 3° caso, calcolo}

$3 + f(n,m) = 2n + m + 4$

= {ip. ind.}

$3 + 2n + m + 1 = 2n + m + 4$

= {calcolo}

$2n + m + 4 = 2n + m + 4$ = {ug} true

Principio di induzione naturale

φ proprietà

$$\Big(\varphi(0) \wedge \big(\forall n \in \mathbb{N}.\ \varphi(n) \Rightarrow \varphi(n+1)\big)\Big)$$

$$\Longrightarrow$$

$$\big(\forall n \in \mathbb{N}.\ \varphi(n)\big)$$

Principio di induzione ben fondata. $(S, <)$ ben fondato

$$\Big(\forall y \in S.\ \underbrace{\big(\forall x \in S.\ x < y \Rightarrow \varphi(x)\big)}_{\text{ip. induttiva}} \Rightarrow \varphi(y)\Big)$$

$$\Longrightarrow \big(\forall y \in S.\ \varphi(y)\big)$$

$$\left(\forall y \in S.\ \left(\forall x \in S.\ x < y \Rightarrow \varphi(x)\right) \Rightarrow \varphi(y)\right)$$

$$\Rightarrow \left(\forall y \in S.\ \varphi(y)\right)$$

se y minimale questa implica si riduce a true. tutto questo

$true \Rightarrow \varphi(y)$

LINGUAGGIO FUNZIONALE

CAML (Camel)

ML Meta Language

era utilizzato per specificare altri linguaggi di programm.

FORMALMENTE

CAML è un linguaggio funzionale

INTERPRETATO

TIPATO

Tutti i valori hanno un unico tipo
(tipo è il dominio dei valori)

CAML è un ling funzionale nel quale si valutano espressioni

```
# 3+4;;
- : int = 7

# true & false;;
- : bool = false

# 'c';;
- : char = 'c'

# "abc";;
- : string = "abc"

# 3.2+.1.3;;
- : float = 4.5
```

```
# let x = 3.5;;
x : float = 3.5

# x +. 1.1;;
- : float = 4.6

# let x = 3;;
x : int = 3
```

~~x := x + 3~~

```
# let f x = x+1;;
```

$$f : \underbrace{\text{int}}_{\text{tipo dell'argomento}} \to \underbrace{\text{int}}_{\text{tipo risultato}} = \langle \text{fun} \rangle$$

```
let g x = x+1+1-1;;
```

```
# let f x = x + 1 ;;
f : int -> int = <fun>

# f 3 ;;
- : int = 4

# (3, 'c') ;;
- : int * char = (3, 'c')
```

```
# let g(n,m) = n + m + 1;;
g : int * int -> int = <fun>

# g(3,4);;
- : int = 8
```

let (f x) y = x + y + 1;;

f si può applicare al parametro x e il risultato della applicazione si può applicare al parametro y

```
# let f x y = x + y + 1;;
f : int -> (int -> int) = <fun>

# f 3 4;;
- : int = 8

# f 3;;
- : int -> int = <fun>
```

\# let g = f 3;;

g : int -> int = <fun>

\# g 4;;

- : int = 8

g = f 3 e (f 3) y = 3 + y + 1

```
let f x y = x + y + 1 ;;
```

<u>Curryed</u>

<u>Curry</u>

let checksum $(x, y, z) = x + y = z$;;

definizione di funzione

uguaglianza

checksum: int * int * int -> bool

= <fun>

```
# checksum (3,4,7);;
- : bool = true
```

```
# checksum (3,4,8);;
- : bool = false
```

$$\text{let cs } \underbrace{x\ \ y\ \ z}_{\text{Curryed}} = x+y=z\ ;;$$

```
cs : int -> int -> int -> bool
   = (fun)
```

```
# let cs x y z = x+y=z ;;
```

$$cs : \underbrace{int}_{\text{tipo di x}} \to \underbrace{int}_{\text{tipo di y}} \to \underbrace{int}_{\text{tipo z}} \to \underbrace{bool}_{\text{tipo risultato}} = \langle fun \rangle$$

```
# let cs1 = cs 3 ;;
cs1 : int -> int -> bool = <fun>
```

```
# cs1 4 8 ;;
- : bool = <false>
```

```
# cs1 4 ;;
- : int -> bool
      = <fun>
```