Titolo nota 24/09/2015

definizione di funzione

$f(n) = n + 2$

$f : \mathbb{Z} \to \mathbb{Z}$

dominio, codominio

n: nome

parametro

$n + 2$: come si ottiene il risultato dell'applicazione (uso) della funzione

$\mathbb{Z}$: insieme degli interi

applicazione della funzione

$f(3)$ — valore del dominio

$= \{$ definizione della f, $n \leftarrow 3 \}$

5

espressione condizionale
(serve per definire le funzioni per casi)

$$f(n) = \begin{cases} \emptyset & \text{se } n < 10 \\ n+1 & \text{altrimenti} \end{cases}$$

$f : \mathbb{Z} \to \mathbb{Z}$

espressione condizionale

$f(n) =$ if $n < 10$ then $\emptyset$ else $n+1$

Titolo nota 24/09/2015

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

$g(n) = n + 3$

sono due nomi diversi

def. di funzioni

$f(g(2), 3)$ ← applicazione di funzioni

$= \{ \text{def. } g, n \leftarrow 2 \}$

$f(5, 3)$

$= \{ \text{def } f, n \leftarrow 5, m \leftarrow 3 \}$

g

Titolo nota 24/09/2015

{ def. di funzioni, applic. di funzioni, espressione condizionale }

no, con solo questi strumenti non si possono risolvere i problemi risolubili dall'informatica.

sì, ci sono problemi che l'informatica non può risolvere. Decidere se un programma terminerà oppure no.

```
while (x>0)
    x = x;
```

Titolo nota 24/09/2015

Ricorsione

definizioni ricorsive di funzione

$f(n) = \text{if } n = 0 \text{ then } 1 \text{ else } 1 + f(n-1)$

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

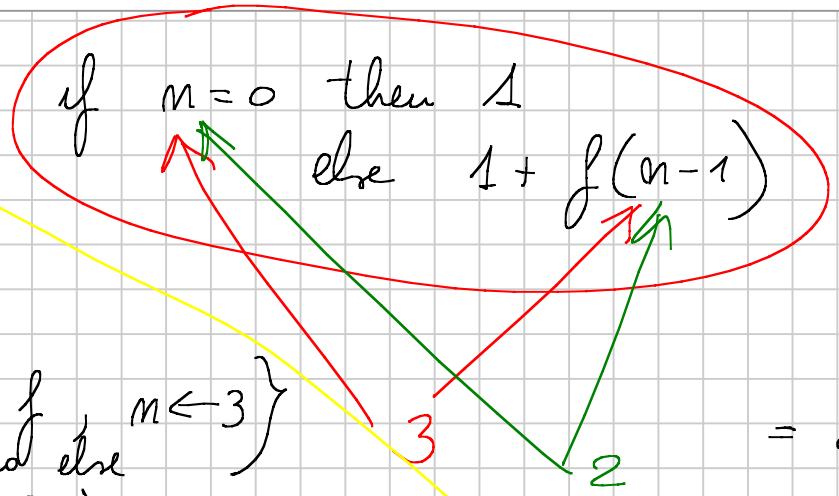
def. ricorsiva di funzione: nella definizione della funzione utilizzo la funzione che sto definendo

Titolo nota 24/09/2015

$f(n) =$ if $n = 0$ then 1 else $1 + f(n-1)$

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

applicazione di funzione = chiamata di funzione



$f(3)$

$= \{$ def f, $n \leftarrow 3$, ramo else $\}$

$1 + f(2)$

$= \{$ def f, $n \leftarrow 2$, ramo else $\}$

$1 + 1 + f(1)$

$= \{$ def f, $n \leftarrow 1$, ramo else $\}$

$1 + 1 + 1 + f(0)$

$= \{$ def f, $n \leftarrow 0$, ramo then $\}$

$1 + 1 + 1 + 1$

$= \{$ calcolo $\}$

4

$f(n) = n + 1$

Titolo nota 24/09/2015

$g(n) =$ if $n=0$ then $\varnothing$ else $g(n+1)$

$g : \mathbb{N} \to \mathbb{N}$

$g(2)$
$=\{$ def g, $n \leftarrow 2$, ramo else $\}$
$g(3)$
$=\{$ def g, $n \leftarrow 3$, ramo else $\}$
$g(4)$
$\vdots$

Titolo nota 24/09/2015

Soluzione del problema MCD

$$MCD(n, n) = n$$

$$MCD(n, m) = MCD(n-m, m) \quad \text{se } n > m$$

$$MCD(n, m) = MCD(n, m-n) \quad \text{se } m > n$$

ricorsiva

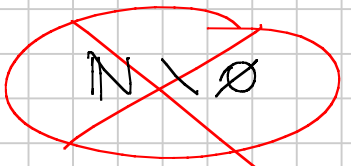
---

def. ricorsiva con espressioni condizionali:

$$mcd(n, m) = \text{if } n = m \text{ then } m \text{ else if } n > m \text{ then } mcd(n-m, m) \text{ else } mcd(n, m-n)$$

$n, m \in \mathbb{N}^+$

$\mathbb{N}^+ = \mathbb{N} \setminus \{\emptyset\}$ ←



definito su insieme

$\mathbb{N}$ insieme

$\emptyset$ non è un insieme

$\{\emptyset\}$ è un insieme che contiene il solo valore $\emptyset$

Titolo nota 24/09/2015

espressione

mcd (18, 15)

= { def mcd, n ← 18, m ← 15 }
ramo else then

mcd (3, 15)

= { def mcd, n ← 3, m ← 15 }
ramo else else

mcd (3, 12)

= { def mcd, n ← 3, m ← 12 }
ramo else else

mcd (3, 9)

= ⋮

mcd (3, 3)

= { def mcd, n ← 3, m ← 3 }
ramo then

3

la valutazione delle espressioni dà come risultato un valore

Titolo nota 24/09/2015

funzione fattoriale (!)

$n!$

$! : \mathbb{N} \to \mathbb{N}$

def. matematica

$$\begin{cases} 0! = 1 \\ n! = n * (n-1) * (n-2) * \dots * 1 \end{cases}$$

ricorsiva

$$\begin{cases} 0! = 1 \\ n! = n * (n-1)! \end{cases}$$

$0!$
$= \{ \text{def } ! \}$
1

$3!$
$= \{ \text{def } ! \}$
$3 * 2 * 1$
$= \{ \text{calcolo} \}$
6

```
fact(n) = if n=0 then 1
          else n * fact(n-1)
```

Titolo nota 24/09/2015

fact (n) = if n = 0 then 1 else n * fact (n-1)

fact (4)

= {def. fact, n ← 4, ramo else}

4 * fact (3)

= {def fact, n ← 3, ramo else}

4 * 3 * fact (2)

= {def fact, n ← 2, ramo else}

4 * 3 * 2 * fact (1)

= {def fact, n ← 1, ramo else}

4 * 3 * 2 * 1 * fact (0)

= {def fact, n ← 0, ramo then}

4 * 3 * 2 * 1 * 1

= {calcolo}

24

per evitare

fact (n) = if n = 0 then 1
else if n = 1 then 1
else n * fact (n-1)

caso ricorsivo

fact (n) =
if n = 1 then 1
else if n = 0 then 1
else n * fact (n-1)

casi base

fact (n) =
if (n = 0) or (n = 1) then 1
else n * fact (n-1)

programmazione imperativa

$\{\langle n, 4\rangle, \langle fact, -\rangle\}$

```
fact = 1;
while (n > 0)
    { fact = fact * n;
      n = n - 1;
    }
```

$\{\langle n, 4\rangle, \langle fact, 1\rangle\}$

$\{\langle n, 3\rangle, \langle fact, 4\rangle\}$

$\{\langle n, 2\rangle, \langle fact, 12\rangle\}$

$\{\langle n, 1\rangle, \langle fact, 24\rangle\}$

$\{\langle n, 0\rangle, \langle fact, 24\rangle\}$

$\{\langle n, 0\rangle, \langle fact, 24\rangle\}$

```
fact = 1;
while (n > 1)
    { fact = fact * n;
      n = n - 1;
    }
```

Titolo nota 24/09/2015

Programmazione imperativa in C (Vedremo)

e funzionale in un linguaggio matematico

(Vedremo la loro traduzione in CAML)

concetti di prog. imperativa e funzionale

che rimarranno gli stessi anche nella programmazione web

Titolo nota 24/09/2015

Sintassi dei linguaggi di programmazione

due metodi per definire la sintassi dei linguaggi di programm.

- automi a stati finiti
- grammatiche a strutture di frase

→ li troverete in molti altri argomenti

→ fa parte di una delle teorie più importanti dell'informatica (teoria dei linguaggi formali)

Titolo nota 24/09/2015

Alfabeto - un insieme finito di **simboli**

$\Lambda = \{a, b\}$

$\Lambda' = \{\rightarrow, \leftarrow, \uparrow, \downarrow\}$ è un alfabeto

abb ≠ bba

stringa su $\Lambda = \{a, b\}$

stringa su $\Lambda = \{a, b\}$

Nei linguaggi formali l'**alfabeto** gioca lo stesso ruolo del **dizionario** nei linguaggi naturali

Stringa su un alfabeto Λ: sequenza (in cui conta l'ordine dei simboli) finita di simboli di Λ

Titolo nota 24/09/2015

$\Sigma = \{a, b\}$

a stringa su Σ di lunghezza 1

aa, ba stringhe su Σ di lunghezza 2

aab " su Σ di lunghezza 3

⋮

ε stringa su Σ di lunghezza $\emptyset$ (stringa vuota)

Σ^+ l'insieme di tutte le stringhe su Σ escluso la stringa vuota

$\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$

Σ^* l'insieme di tutte le stringhe su Σ

$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \ldots\}$ infinito