

Definire un'applicazione

Nel primo volume di "Fare Matematica" si legge:

"Un'applicazione tra due insiemi è un *azione* che porta da un insieme a un altro.

Essa consiste di:

- un *insieme non vuoto* X detto **dominio** dell'applicazione;
- un *insieme non vuoto* Y detto **codominio** dell'applicazione;
- una **regola** r , che assegna ad ogni elemento x del dominio X esattamente un elemento y del codominio Y ."

In Maple è possibile definire un oggetto con il comando "`:=`" che consente di assegnare a una "variabile" o "etichetta" un determinato valore. Con lo stesso comando si possono definire le funzioni;

$$f := x \rightarrow f(x) :$$

La notazione, evidenzia la regola, cioè l'insieme delle operazioni da eseguire sull'*argomento* per ottenere l'*immagine*.

Osservazione. Il punto e virgola ";" termina un comando e ne mostra il risultato dopo aver premuto il tasto Return; i due punti ":" sono utili per istruzioni che non richiedono una visualizzazione immediata del risultato, come l'assegnazione di una variabile, di un'espressione o di una funzione. Entrambi i segni sono necessari per separare le istruzioni all'interno di blocchi di codice più complessi come *if-then-else*, cicli *for*, *while-do* o *procedure*.

Esempi di assegnazione:

$$f := x \rightarrow f(x) :$$

$$g := x \rightarrow \frac{\ln(x)}{1 + \ln(x)} :$$

E' indispensabile fare presente che nella definizione di applicazione, la regola è basata solo sul legame che deve intercorrere fra la variabile x e la variabile y :

ad ogni elemento x del dominio X corrisponde esattamente uno ed un solo elemento y del codominio Y

Quindi, non si richiede affatto che per una funzione questo legame abbia in tutto il suo dominio la medesima rappresentazione analitica.

Ad esempio, si può definire la seguente regola:

$$f := \begin{cases} 1 - x & x \leq 0 \\ e^{-x} & 0 < x \end{cases}$$

Una funzione soddisfacente tali condizioni può essere assegnata nel modo seguente:

```
f := proc(x)
if x ≤ 0 then 1 - x
elif 0 < x then e-x
fi
end:
```

Per ottenere il diagramma del grafico di una funzione si adopera il comando *plot*. Se la funzione è definita da rappresentazioni analitiche diverse si ricorre al comando *plot* ('funzione', 'x'=a..b), dove gli apici sono obbligatori.

Il diagramma del grafico della funzione **(3)** limitato si ottiene con il comando:

```
plot('f(x)', 'x'=-∞..∞)
```

