

Frecce dritte e frecce storte

Giulio Canti

April 19, 2018

Cos'è la programmazione funzionale?

La programmazione funzionale è programmare con le funzioni
(fate conto che abbia inserito qui un meme appropriato)

Programmi senza effetti

$$f : A \rightarrow B$$

La funzione f modella un programma con un input di tipo A e che produce un output di tipo B

Example

```
function len(s: string): number {  
    return s.length  
}
```

Come funzione:

$$\text{len} : A \rightarrow B$$

- $A = \text{string}$ (input)
- $B = \text{number}$ (output)

$$f : A \rightarrow M\langle B \rangle$$

La funzione f modella un programma con un input di tipo A e che produce un output di tipo B insieme ad un effetto di tipo M

Example

```
function head(as: Array<string>): Option<string> {  
  return as.length == 0 ? none : some(as[0])  
}
```

Come funzione:

$$\text{head} : A \rightarrow M\langle B \rangle$$

- $A = \text{Array}\langle \text{string} \rangle$ (input)
- $B = \text{string}$ (output)
- $M = \text{Option}$ (effetto)

Composition is the essence of programming - Bartosz Milewski

Come si compongono i programmi?

Come si compongono i ~~programmi~~ **le funzioni**?

Data la funzione g

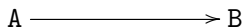
$$g : A \rightarrow B$$

e la funzione f

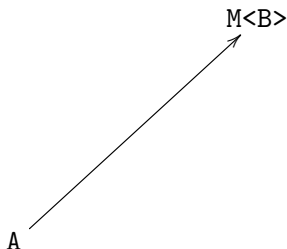
$$f : C \rightarrow D$$

quando è possibile comporre?

I programmi senza effetti sono rappresentati da **frecce dritte**



I programmi con effetti sono rappresentati da **frecce storte**

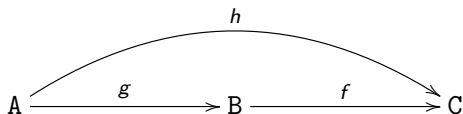


Due programmi senza effetti

$$A \xrightarrow{g} B \xrightarrow{f} D$$

Condizione: $\text{Output}(g) = \text{Input}(f)$

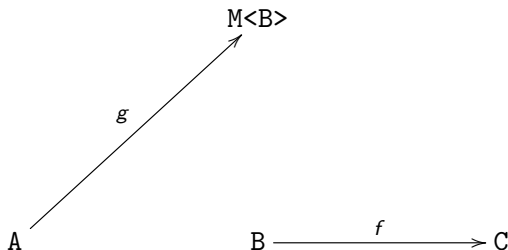
Due programmi senza effetti



Soluzione

$$h = f \circ g$$

Un programma con effetti + un programma senza effetti

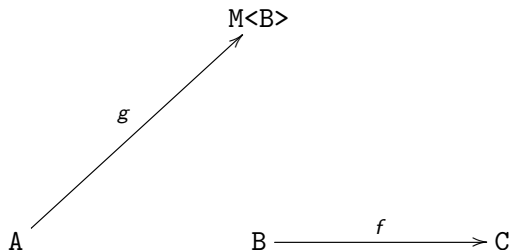


Condizione: $\text{Output}(g) = M\langle \text{Input}(f) \rangle$

```
interface Functor<M> {  
    map: <A, B>(f: (a: A) => B) => (ma: M<A>) => M<B>  
}
```

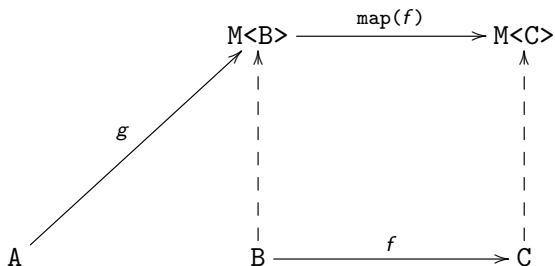
```
interface Functor<M> {  
    map: <A, B>(f: (a: A) => B) => ( (ma: M<A>) => M<B> )  
}
```


Un programma con effetti + un programma senza effetti

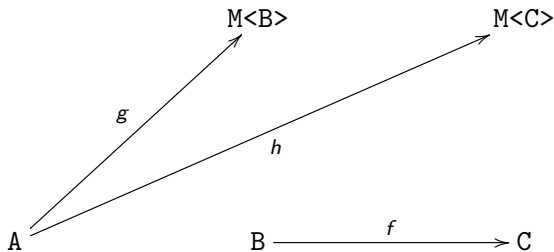


Condizione: M ammette una istanza di Functor

Un programma con effetti + un programma senza effetti



Un programma con effetti + un programma senza effetti



Soluzione

$$h = \text{map}(f) \circ g$$

$$f : A \times B \rightarrow C$$

Currying

$$f' : A \rightarrow B \rightarrow C$$

Funzioni con due parametri

Example

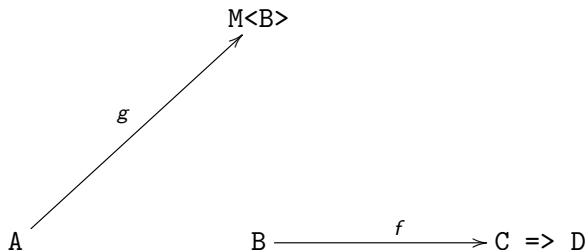
```
const sum = (a: number, b: number): number => a + b
```

Currying

Example

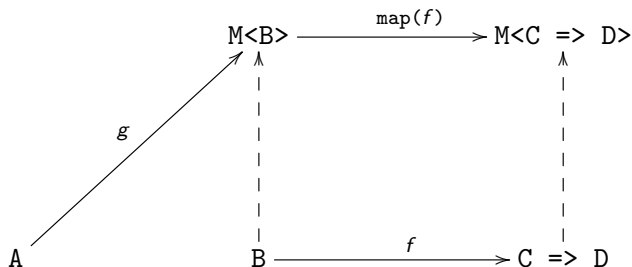
```
const sum = (a: number) => (b: number): number => a + b
```

Un programma con effetti + un programma senza effetti con due parametri



Condizione: $\text{Output}(g) = M\langle \text{Input}(f) \rangle$

Un programma con effetti + un programma senza effetti con due parametri



Condizione: M ammette una istanza di Functor

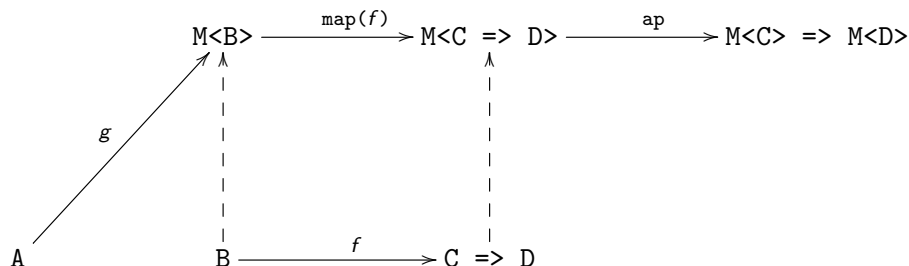
Applicative

```
interface Applicative<M> extends Functor<M> {  
  of: <A>(a: A) => M<A>,  
  ap: <A, B>(mab: M<(a: A) => B>) => (ma: M<A>) => M<B>  
}
```


Applicative

```
interface Applicative<M> extends Functor<M> {  
  of: <A>(a: A) => M<A>,  
  ap: <A, B>(mab: M<(a: A) => B>) => ( (ma: M<A>) => M<B> )  
}
```

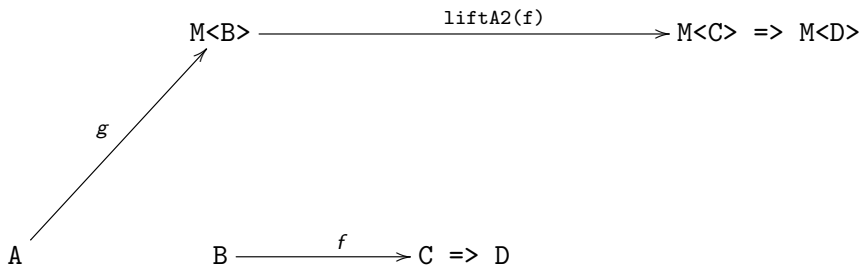
Un programma con effetti + un programma senza effetti con due parametri



Condizione: M ammette una istanza di `Applicative`

Composizione

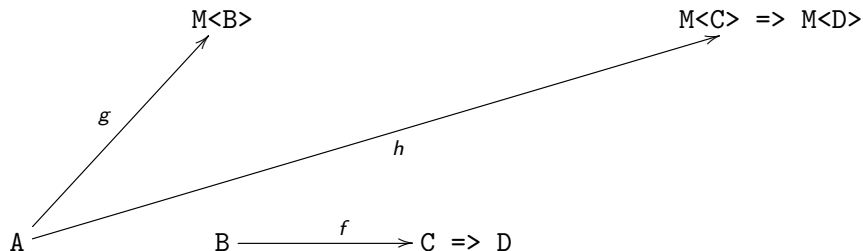
Un programma con effetti + un programma senza effetti con due parametri



Ove

$$\text{liftA2} = \text{ap} \circ \text{map}$$

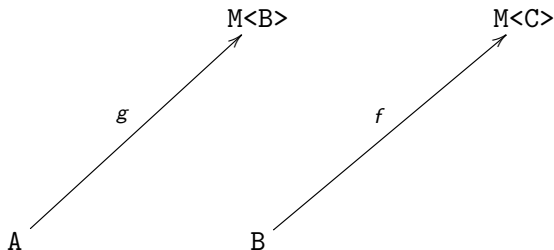
Un programma con effetti + un programma senza effetti con due parametri



Soluzione

$$h = \text{liftA2}(f) \circ g$$

Un programma con effetti + un programma con effetti

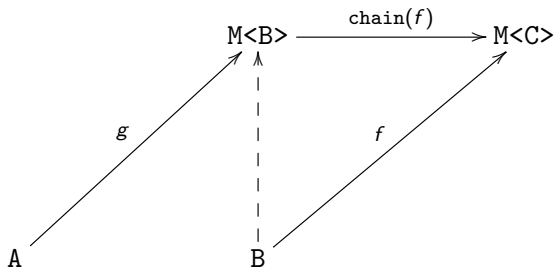


Condizione: $\text{Output}(g) = M\langle \text{Input}(f) \rangle$

```
interface Monad<M> extends Functor<M> {  
    of: <A>(a: A) => M<A>,  
    chain: <A, B>(f: (a: A) => M<B>) => (ma: M<A>) => M<B>  
}
```

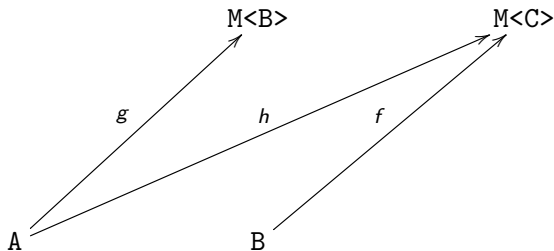
```
interface Monad<M> extends Functor<M> {  
    of: <A>(a: A) => M<A>,  
    chain: <A, B>(f: (a: A) => M<B>) => ( (ma: M<A>) => M<B> )  
}
```

Un programma con effetti + un programma con effetti



Condizione: M ammette una istanza di Monad

Un programma con effetti + un programma con effetti



Soluzione

$$h = \text{chain}(f) \circ g$$