Proprietà per numeri interi

Problemi

- date una funzione e una specifica di ciò che la funzione dovrebbe calcolare, la funzione è corretta rispetto alla specifica?
- date due funzioni (per es. una ovviamente corretta ma inefficiente, l'altra efficiente ma più complessa da capire), sono equivalenti?

Vari approcci, tra cui:

- ► Test
 - più facile (specialmente se il linguaggio non è imperativo)
 - analisi non esaustiva (possono esserci errori in casi non considerati)
- ▶ Dimostrazione
 - più difficile (specialmente se il linguaggio è imperativo)
 - analisi esaustiva

Il primo approccio, quello dei **test**, è fattibile solo se ci sono un numero finito di casi.

Tuttavia, nella maggior parte delle volte non è così e quindi un approccio basato sui test non è esaustivo.

Dobbiamo dunque passare ad un approccio più matematico e rigoroso:la dimostrazione.

principio di induzione sui numeri naturali

Data una proprietà P(n) dei numeri naturali, se

- **▶** *P*(0) e
- ▶ P(n) implica P(n + 1) per ogni $n \in \mathbb{N}$, allora P(n) per ogni $n \in \mathbb{N}$.

Esponenziale

```
exp :: Int \rightarrow Int \rightarrow Int
exp _ 0 = 1
exp x n = x * exp x (n - 1)
```

Di questa funzione vorremmo verificare alcune proprietà:

- 1 $\forall x, m \geq 0, n \geq 0 : \exp x (m+n) = \exp x m * \exp x n$

Esempio prima proprietà: $2^{(5+3)} = 2^5 * 2^3$

Esempio seconda proprietà: $(2*2)^5 = 2^5*2^5$

Dimostrazione prima proprietà

Passo base:



Essendo 0 l'elemento neutro della somma, $exp\ x\ (0+n)$ si riduce a $exp\ x\ n.$

Essendo 1 l'elementro neutro della moltiplicazione, possiamo trasformare $exp \ x \ n$ in $1 * exp \ x \ n$.

Ma sappiamo per definizione della funzione exp che $1=\exp x\ 0$ quindi arriviamo alla conclusione che $\exp x\ (0+n)=\exp x\ 0*\exp x\ n$

Passo induttivo:

$$P(m-1) = P(m) \quad \forall m > 0$$
 $exp \quad x \quad (m+m)$
 $= x * exp \quad x \quad ((m+n)-1) \qquad exp. 2$
 $= x * exp \quad x \quad ((m-1)+m) \qquad prop. dite = = x * (exp \quad x \quad (m-1) * exp \quad x \quad ip. ind.$
 $= (x * exp \quad x \quad (m-1)) * exp \quad x \quad order. di *$
 $= exp \quad x \quad m \quad * exp \quad x \quad m \quad exp. 2$

Per definizione della funzione, sappiamo che $\exp x\ (m+n)=x*\exp x\ ((m+n)-1)=x*\exp x\ ((m-1)+n).$ Per ipotesi induttiva $(x^{m+n}=x^n*x^m$ è vera per P(m-1)) sappiamo che

Tuttavia la prima parte coincide con la definizione della funzione $\exp x n = x * \exp x (n - 1)_{e \ la \ possiamo}$ dunque riscrivere in $exp \ x \ n$ e otteniamo quindi $exp \ x \ m * exp \ x \ n$

Dimostrazione seconda proprietà

Fattela a casa.

Proprietà per liste

Come per i numeri interi, utilizziamo un metodo induttivo per verificare le proprietà.

Tuttavia, il **passo base** e il **passo induttivo** devono adattarsi al tipo della **lista**:

principio di induzione sulle liste finite

Data una proprietà P(xs) delle liste, se

- **▶** *P*([]) e
- ightharpoonup P(xs) implica P(x:xs) per ogni x e xs, allora P(xs) per ogni lista finita xs.

Analogia con i numeri interi:

- P(0) diventa P([])
- $P(n) \Rightarrow P(n+1)$ diventa $P(xs) \Rightarrow P(x:xs)$
 - xs è la lista di partenza(n) e facendo l'append (:) di x, otteniamo una lista più lunga(n+1)

Funzioni notevoli sulle liste

```
length :: [a] \rightarrow Int
length [] = 0
length (_ : xs) = 1 + length xs
```

```
(++) :: [a] \rightarrow [a] \rightarrow [a] (++) [] ys = ys (++) (x : xs) ys = x : (++) xs ys
```

```
reverse :: [a] \rightarrow [a]
reverse [] = []
reverse (x : xs) = reverse xs ++ [x]
```

Proviamo a dimostrare la prima delle seguenti proprietà:

1
$$\forall xs, ys : length(xs ++ ys) = length xs + length ys$$

2
$$\forall xs, ys, zs : xs ++ (ys ++ zs) = (xs ++ ys) ++ zs$$

$$\forall xs, ys : reverse(xs ++ ys) = reverse ys ++ reverse xs$$

Sorge subito un problema: su quale lista facciamo induzione, xs o ys?

Una regola informale stabilisce che **conviene** fare induzione sulla lista che viene "analizzata di più" dalle funzioni:

Prendendo in esame la **prima proprietà**, notiamo che in length(xs++ys), xs viene analizzata una volta da (++), mentre ys 0 volte.

Infine length analizza una volta la lista risultante dalla concatenazione delle due liste.

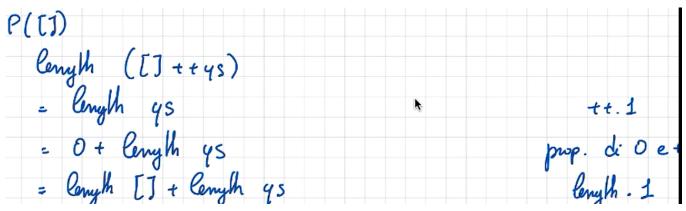
Dall'altro lato, vengono analizzate entrambe le liste una volta:

length
$$(xs ++ ys) = length xs + length ys$$

Scelgo dunque la lista xs.

Dimostrazione

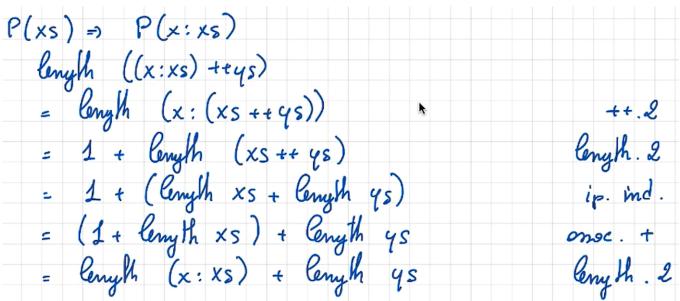
Passo base:



Molto intuitivo: sappiamo che la concatenazione di [] a ys restituisce ys stesso. Ciò equivale a $lenght\ ys + 0$ (lo 0 è l'elemento neutro della somma).

Lo 0 è anche dato da length[] quindi sono arrivato che $length([]++ys)=length\ xs\ ys$

Passo induttivo:



Nel primo passaggio, uso la seconda equazione di (++) per

Utilizzo poi la seconda equazione di length per arrivare a

Utilizzando l'associatività della somma e applicando anche la seconda equazione di length(questa volta da destra verso sinistra, arrivo alla conclusione della dimostrazione.

Nota bene: Il principio di induzione si basa sul fatto che, se vale P(n), allora vale anche P(n+1).

Dunque, se durante la mia dimostrazione, arrivo ad avere P(n), allora lo posso sostituire con P(n+1)