Ricorsione, Principio di Induzione e Correttezza (parziale)

La progettazione di un algoritmo ricorsivo `R` covariante ripercorre fedelmente il ***Principio di Induzione*** sul quale si basa la dimostrazione che `R` è *parzialmente corretto*.

Algoritmo `R` parzialmente corretto? Supponiamo che l'algoritmo `R` dipenda da un parametro formale `x` che può assumere valori nell'insieme dei numeri naturali.

Si dice che "`R` è parzialmente corretto" quando, "per ogni valore `n` assegnato ad `x`, `R(n)` produce il risultato atteso".

Come si dimostra che `R` è (parzialmente) corretto? Applicando il Principio di Induzione, **come abbiamo fatto sinora** durante la progettazione di algoritmi sia ricorsivi, sia iterativi, anche se, in questo secondo caso, in maniera meno evidente.

Applicare il Principio di Induzione consiste nel percorrere i seguenti passi:

1. Si individua un predicato `P`, cioè una proprietà che descrive il comportamento di `R`
2. Si dimostra la verità di un predicato, cioè di un'affermazione, che descrive il comportamento di `R` nel *caso base*, cioè quello più semplice:

   "`R(0)` è corretto se produce un risultato per cui `P(0)` è un'affermazione vera".

   Esempi di predicati che descrivono il comportamento di algoritmi nel *caso base*:

   - `mol(x,0) == 0`.

     Formalmente, cioè usando il linguaggio della logica matematica, si scrive `P(0)` che, nel caso in esame è il *predicato* `mol(x,0) == 0`, cioè un'affermazione che può essere vera o falsa.
   - `meno(x,0) == x`.

     Formalmente, cioè usando il linguaggio della logica matematica, si scrive `P(0)` che, nel caso in esame, è il *predicato* `meno(x,0) == x`, cioè un'affermazione che può essere vera o falsa.
   - `esp(x,0) == 1`.

     Formalmente, cioè usando il linguaggio della logica matematica, si scrive `P(0)` che, nel caso in esame, è il *predicato* `esp(x,0) == 1`, cioè un'affermazione che può essere vera o falsa.
3. Si dimostra la verità di un predicato che descrive il comportamento di `R` nel *caso induttivo*, cioè nel caso in cui, assumendo che `R` si sia comportato

secondo le aspettative sino al passo precedente, `R` si comporta secondo le attese per un ulteriore passo:

> "Per un qualsiasi valore `n`, se `R(n-1)` è corretto, cioè produce un risultato che rende vero `P(n-1)`, allora riusciamo a spiegare senza ombra di dubbio, cioè a *dimostrare*, che anche `R(n)` è corretto, cioè che produce un risultato che rende vero `P(n)`".

Esempi di predicati che descrivono il comportamento di un algoritmo nel *caso induttivo*:

- per un qualsiasi valore `n`, se `mol(x,n-1) == x*(n-1)` è vero, allora `mol(x,n) == x*n` è vero.

  Formalmente, cioè usando il linguaggio della logica matematica, si scrive $\forall$ `n > 0, P(n-1)`$\rightarrow$`P(n)`, in cui `P(n)` è il *predicato* `mol(x,n) == x*n`, cioè un'affermazione che può essere vera o falsa;

- per un qualsiasi valore `n`, se `meno(x,n-1) == x-(n-1)` è vero, allora `meno(x,n) == x-n` è vero.

  Formalmente, cioè usando il linguaggio della logica matematica, si scrive $\forall$ `n, P(n-1)`$\rightarrow$`P(n)`, in cui `P(n)` è il *predicato* `meno(x,n) == x-n`, cioè un'affermazione che può essere vera o falsa;

- per un qualsiasi valore `n`, se `esp(b,n-1) == b^{n-1}` è vero, allora `esp(b,n) == b^{n}` è vero.

  Formalmente, cioè usando il linguaggio della logica matematica, si scrive $\forall$ `n, P(n-1)`$\rightarrow$`P(n)`, in cui `P(n)` è il *predicato* `esp(b,n) == b^{n}`, cioè un'affermazione che può essere vera o falsa.

Principio di Induzione

Portare a termine i passi descritti nella sezione precedente, equivale ad applicare il Principio di Induzione che giustifica la validità di un predicato `P(n)` per un qualsiasi numero naturale `n`.

Un po' più formalmente, dato un predicato `P(n)`, in cui `n`$\in \mathbb{N}$:

- se dimostriamo `P(0)`, e
- se dimostriamo che, per ogni `n > 0`, l'implicazione `P(n-1)`$\rightarrow$ `P(n)` è vera,

allora per il *Principio di Induzione* logica possiamo concludere che per ogni `n`, il predicato `P(n)`è vero.

In formule logiche, quindi ancora più formalmente, si scrive `P(0)`$\wedge$ ($\forall$`n > 0, P(n-1)`$\rightarrow$`P(n)`)$\rightarrow$($\forall$`n`$\in \mathbb{N}$, `P(n)`).

Esempi del Principio di Induzione che abbiamo (implicitamente) usato per le dimostrazioni di correttezza parziale sono:

1. `(mol(x,0)==0)`$\wedge$`(`$\forall$`n > 0`, `(mol(x,n-1)==x*(n-1))`$\rightarrow$`(mol(x,n)==x*n))`$\rightarrow$`(`$\forall$`n`$\in \mathbb{N}$, `mol(x,n)==x*n)`.
2. `(meno(x,0)==x)`$\wedge$`(`$\forall$`n > 0`, `(meno(x,n-1)==x-(n-1))`$\rightarrow$`(meno(x,n)==x-n))`$\rightarrow$`(`$\forall$`n`$\in \mathbb{N}$, `meno(x,n)==x-n)`.
3. `(esp(x,0)==1)`$\wedge$`(`$\forall$`n > 0`, `(esp(x,n-1)==x^(n-1))`$\rightarrow$`(esp(x,n)==x^n))`$\rightarrow$`(`$\forall$`n`$\in \mathbb{N}$, `esp(x,n)==x^n)`.

Wikipedia. Ovviamente esistono siti su cui leggere e rileggere del principio di induzione. UNo ragionevole è Principio di Induzione (su Wikipedia, per esempio). Non è in generale abbondandte il materiale che associa molti esempi sull'uso del Principio di Induzione per provare la correttezza parziale di un algoritmo.