doppio $\left[\begin{array}{l} 2i = d \qquad\qquad i' = i + 1 \\ d' = d + 2 = 2i + 2 = (i+1) \cdot 2 = i' \cdot 2 \end{array}\right.$

Introduzione alla correttezza dei programmi
— Il Principio di Induzione —

Il Principio di Induzione

Quando abbiamo una proposizione P (una frase affermativa con un valore di verità definito), possiamo creare un predicato P(x) rimpiazzando un nome (di individuo) che compare nella proposizione con una variabile x. Per esempio:

P = Cesare ha varcato il Rubicone ⇝ P(x) = x ha varcato il Rubicone
$Q = 3^2 > 2^2 \rightsquigarrow Q(x) = x^2 > 2^2$

In entrambi questi esempi P(x) è una proposizione vera o falsa secondo i valori della variabile x:

P(Cesare) è una proposizione vera, in realtà coincide con la proposizione P, mentre P(Trump) è falsa. Analogamente Q(3) è vera mentre Q(2) è falsa.

Quando un predicato viene derivato in questo modo da una proposizione, la variabile che prende il posto del nome varia su un insieme fissato di possibili valori: nome di persone nel primo caso, nomi di numeri nel secondo.

Diciamo che un predicato P(x) è relativo ai numeri naturali se forma proposizioni quando la variabile x viene rimpiazzata da espressioni numeriche che hanno come valori numeri naturali. Per esempio, Q(x) è un predicato relativo ai naturali se ci restringiamo ai valori naturali di x: non è vero di ogni numero naturale, ma ogni proposizione Q(n) è vera o falsa secondo il valore n.

Il Principio di Induzione

Per dimostrare che una proprietà è vera di ogni numero naturale, sono disponibili alcune tecniche.

Per esempio, se P(x) = il quadrato del numero 2x è divisibile per 4, possiamo dimostrare che $\forall x\, P(x)$ semplicemente osservando che , per ogni numero naturale x, $(2x)^2 = 4x^2$, che è chiaramente divisibile per 4.

In altri casi, occorre invece riuscire a trovare una descrizione finita di un numero infinito di inferenze della forma

$$
\dfrac{\dfrac{\dfrac{P(0) \qquad P(0) \longrightarrow P(1)}{P(1)} \qquad P(1) \longrightarrow P(2)}{P(2)} \qquad P(2) \longrightarrow P(3)}{P(3)} \qquad \vdots
$$

Il Principio di Induzione è un modo di riassumere in un solo schema un numero infinito di inferenze come questo.

Il Principio di Induzione

Sia P(n) un predicato relativo ai numero naturali, $n \geq 0$. Supponiamo che:

- (**Base dell'induzione**) P(0) sia vera, e che
- (**Passo induttivo**) assumendo che P(k) sia vera, si riesca a dimostrare che P(k+1) è vera.

Il **principio di induzione** afferma che da queste premesse segue che

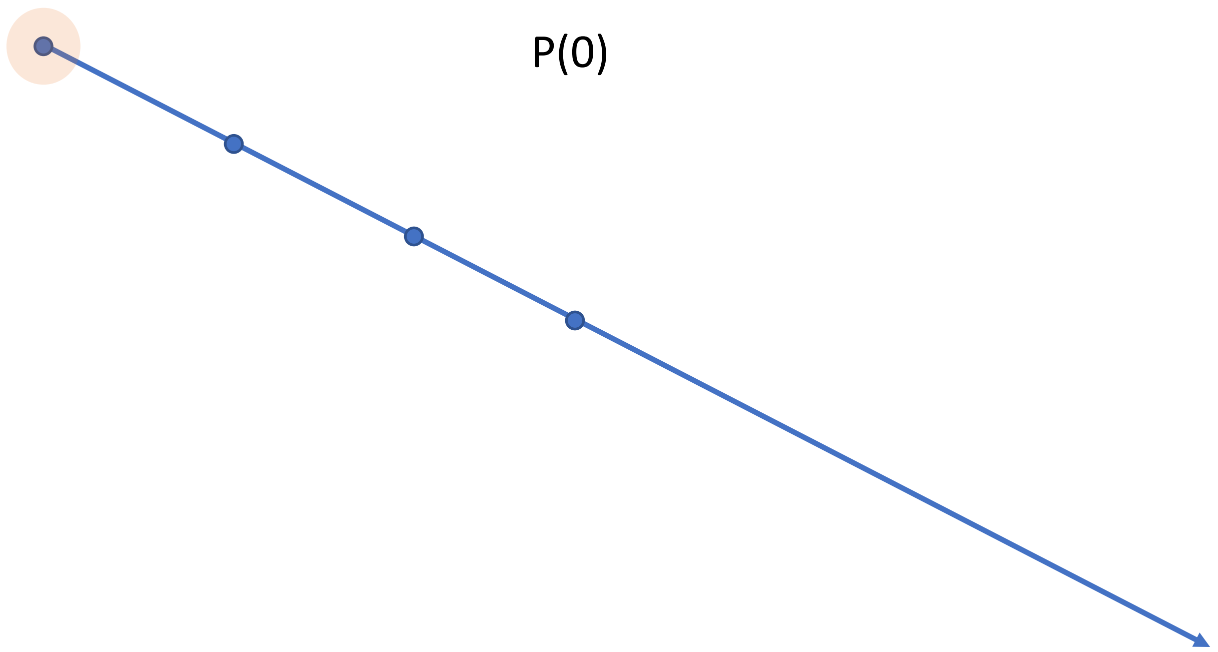
- P(n) è vera per tutti i valori naturali di n.

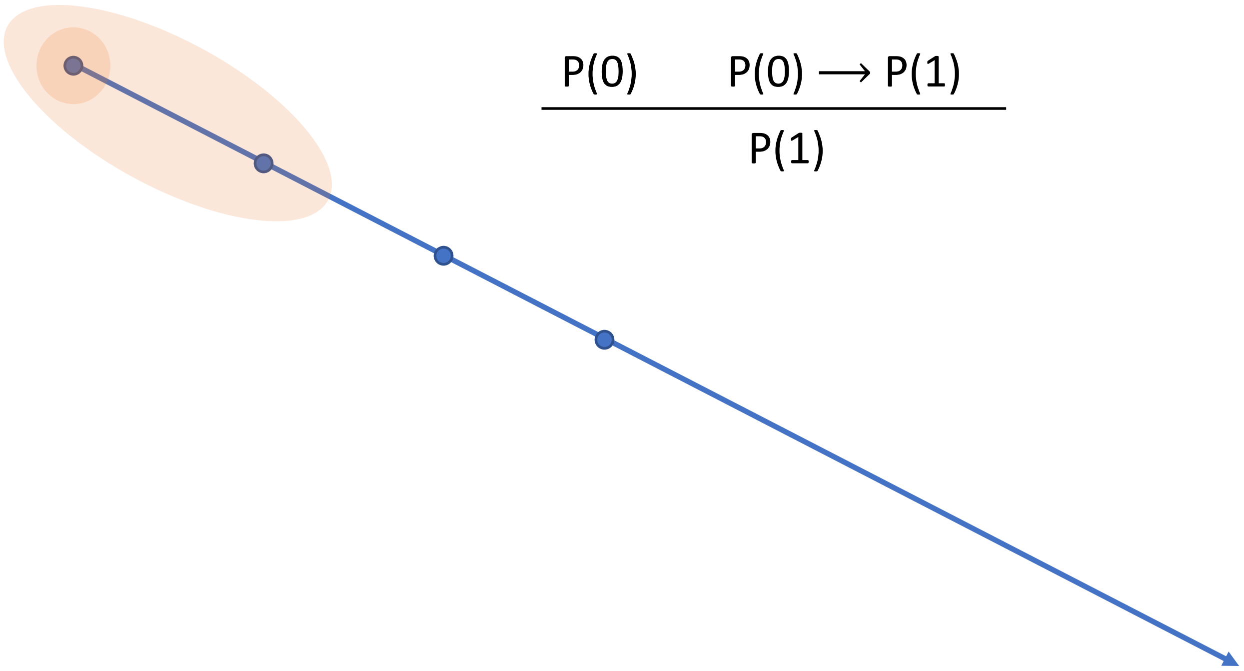
Una giustificazione (intuitiva) del principio di induzione si può avere dall'osservazione che per ogni numero naturale n:

- $n = 0$, oppure
- $n > 0$, quindi $n = k+1$ per un unico k.

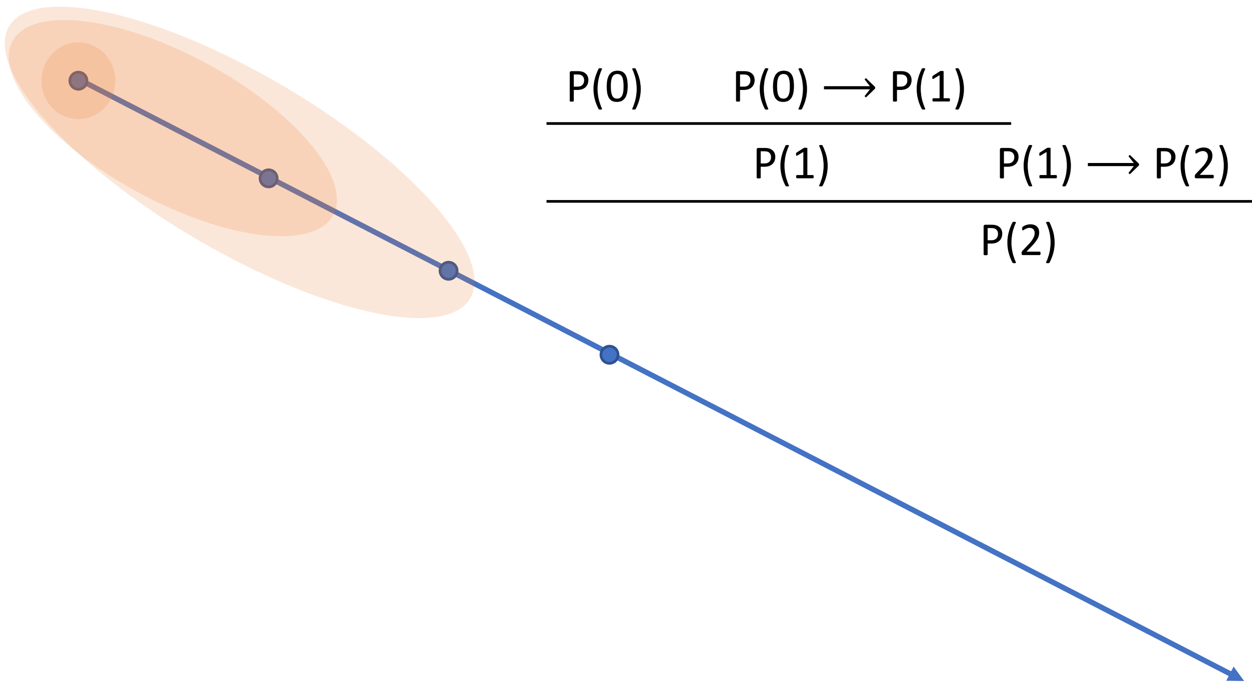
La base dell'induzione ed il passo induttivo permettono allora di generare la serie infinita di inferenze che consentono di dimostrare P(k) per ogni specifico valore naturale di k.

Con un disegno:

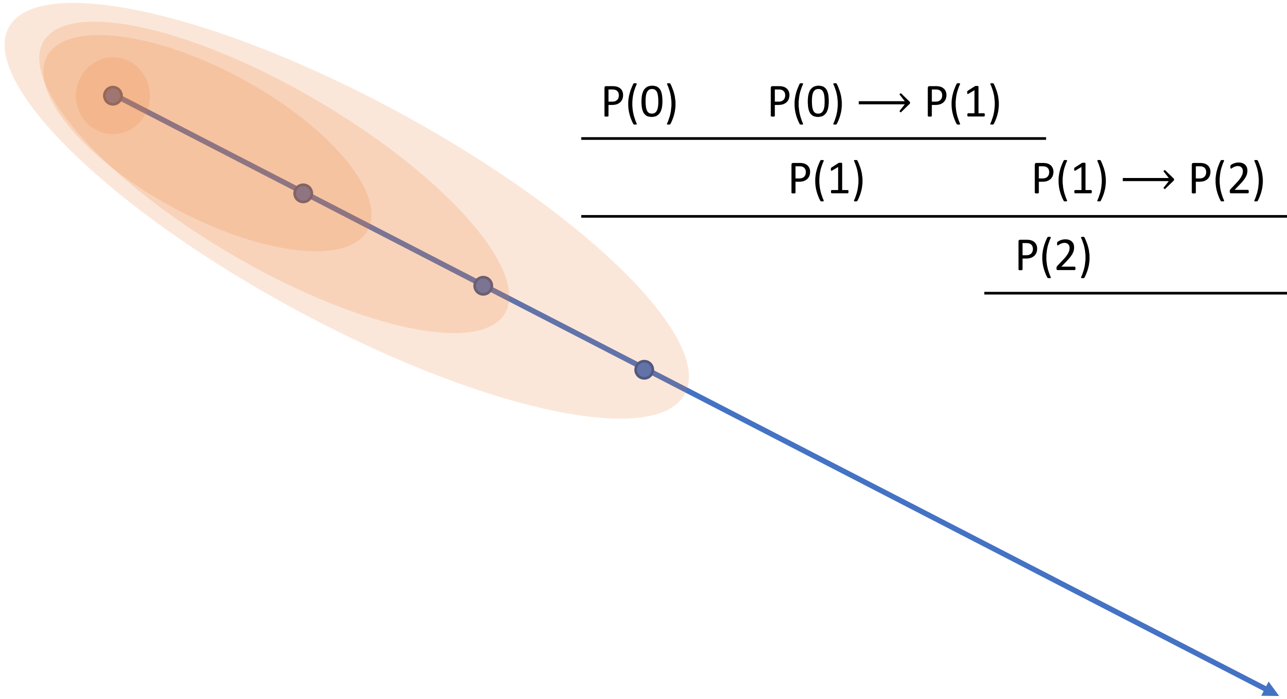

P(0)



$$\frac{P(0) \qquad P(0) \longrightarrow P(1)}{P(1)}$$



$$\dfrac{\dfrac{P(0) \qquad P(0) \longrightarrow P(1)}{P(1)} \qquad P(1) \longrightarrow P(2)}{P(2)}$$



$$\dfrac{\dfrac{\dfrac{P(0) \qquad P(0) \longrightarrow P(1)}{P(1)} \qquad P(1) \longrightarrow P(2)}{P(2)} \qquad P(2) \longrightarrow P(3)}{P(3)} \qquad \vdots$$

Usare il principio di induzione per dimostrare la correttezza di programmi tramite proprietà invarianti

Vediamo come il principio di induzione ci guida nella dimostrazione di correttezza del codice per il calcolo del fattoriale:

```
int x = 0;
int y = 1;
while (x < n) {  //invariante: y = x!
   x = x + 1;
   y = x * y;
}
```

Per dimostrare l'invarianza del predicato , occorre dimostrare che

$$\forall k \geq 0 \; P(k)$$

dove

P(k) = dopo la k-esima iterazione, `y = x!`

Base: P(0), cioè dopo la 0-esima iterazione, `y = x!`, cioè prima dell'esecuzione del ciclo `y = x!`.
Infatti `y` = 1 = 0! = `x!`

Passo induttivo:

```
while (x < n) {
    x = x + 1;
    y = x * y;
}
```

Prendiamo un generico k ≥ 0. **Assumiamo** che:

P(k), cioè
dopo la k-esima iterazione, $\boxed{y = x!}$

allora:

dopo l'iterazione successiva (la k+1-esima) abbiamo

$$
\begin{aligned}
x' &= x + 1 \\
y' &= x' * y = (x + 1) * \boxed{y} \\
&= (x + 1) * \boxed{x!} \\
&= (x + 1)!
\end{aligned}
$$

per ipotesi induttiva

quindi: P(k+1)

Passo induttivo:

```
while (x < n) {
    x = x + 1;
    y = x * y;
}
```

Prendiamo un generico k ≥ 0. **Assumiamo** che:

P(k), cioè
dopo la k-esima iterazione, $\boxed{y = x!}$

per ipotesi induttiva

allora:

dopo l'iterazione successiva (la k+1-esima) abbiamo

$$
\begin{aligned}
x' &= x + 1 \\
y' &= x' * y = (x + 1) * \boxed{y} \\
&= (x + 1) * \boxed{x!} \\
&= (x + 1)!
\end{aligned}
$$

quindi: P(k+1)

Quindi, per il Principio di Induzione: ∀k ≥ 0 P(k)
Questo significa che la proprietà **`y = x!`** è **invariante**.
La **correttezza** del programma segue dal fatto che, all'uscita dal ciclo, `x = n`, quindi per l'invariante `y = x! = n!`