# O Princípio da Indução e suas Aplicações

Felipe Luís Pinheiro*

18 de novembro de 2018

**Resumo**

Neste trabalho mostraremos diversas utilizações do princípio de indução em Ciência da Computação.

Começamos discutindo o principio da indução e suas implicações, posteriormente definimos os problemas a serem discutidos e por fim discutimos os problemas propostos até a sua completa solução.

## 1 Introdução

Principio da Indução finita **(PIM)** é formalmente definido como:
**Definição 1**. Seja $P(n)$ uma quantidade associada aos números naturais. $P$ é um predicado unário cujo argumento é um número natura, e suponhamos que:

**Base de Indução (BI)** : $p(1)$ é verdadeiro;

**Passo Indutivo (PI)** : para todo número natura $n$, se $P(n)$ é verdadeiro então $P(n+1)$ é verdadeiro

Nesta condição, a propriedade $p(n)$ é verdadeira para todo número natural $n$.

Para se realizar uma prova mediante processo indutivo devemos provar que **BI** é verdadeiro, o que pode ser provado mediante aplicação direta da propriedade em estudo, e após devemos realizar o **PI** e provar que a propriedade continua válida para $n+1$ qualquer, considerando que ela é válida pra $n$.

também pode ocorrer de existir uma propriedade que só comece a ser válida depois de algum valor específico, sendo assim definimos o principio da indução finita generalizada **(PIMG)** como:
**Definição 2**: Seja $p(n)$ uma propriedade sobre os números naturais que satisfaz as seguintes condições:

**(BI)** : O número natural $m_0$ satisfaz a propriedade $p$;

**(PI)** : Se um número natural $n$ satisfaz a propriedade $P$ então seu sucessor também satisfaz a propriedade $P$.

Então todos os números naturais maiores ou iguais a $m_0$ satisfazem a propriedade $P$.

Para finalizar precisamos definir o principio da indução forte **(PIF)**, como:
**Teorema 1**: Seja $P$ uma propriedade referente aos números naturais. Dado $n \in \mathbb{N}$, se a validade de $P$ para todo número natural menor do que $n$ implicar que $P$ é verdadeiro para $n$, então $P$ é verdadeira para todos os números naturais. Ou ainda, se $\forall_n(\forall_m, m < n \to P(m)) \to p(n)$ então $\forall_n, P(n)$.

*matricula:18/0052667

# 2 Indução Estrutural

# 3 Correção

# 4 Problemas

Nesta seção mostramos os dois problemas a serem discutidos e também mostramos as suas respectivas soluções.

## 4.1 Problema 1

**Problema 1**: Prove a equivalência entre os princípios da indução forte (PIF) e da indução matemática (PIM)[1].

Precisamos provar que PIF $\Leftrightarrow$ PIM, ou seja, PIF $\rightarrow$ PIM e PIM $\rightarrow$ PIF, para fazer isso queremos escrever tanto o PIF como o PIM como sequentes de lógica indutiva e fazer a arvore de demonstração.

Desta forma podemos escrever o PIFG como sequente da seguinte forma

$$\exists_m \psi \rightarrow \forall_n \psi, \text{ com } n > m \tag{1}$$

e o PIM como

$$\forall_m \psi \rightarrow \forall_n \psi, \text{ com } n > m. \tag{2}$$

Em ambos os caso vemos que $n > m$, logo é um cuidado extra que temos que tomar durante a nossa demonstração. Seguimos a abaixo com a demonstração das arvores de prova.

**PIFG $\rightarrow$ PIM**

Nosso objetivo agora é provar a implicação PIF $\rightarrow$ PIM, ou seja, o sequente

$$\exists_m \psi \rightarrow \forall_n \psi \vdash \forall_m \psi \rightarrow \forall_n \psi \tag{3}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[\forall_m \psi]^a}{\psi[m/x]}\,(\forall_e)}{\exists_m \psi}\,(\exists_i) \qquad \exists_m \psi \rightarrow \forall_n \psi}{\forall_n \psi}\,(\rightarrow_e)}{\forall_m \psi \rightarrow \forall_n \psi}\,(\rightarrow_i)^a$$

**PIM $\rightarrow$ PIF**

Agora queremos provar o sentido inverso, ou seja, a implicação PIM $\rightarrow$ PIF ou o sequente

$$\forall_m \psi \rightarrow \forall_n \psi \vdash \exists_m \psi \rightarrow \forall_n \psi \tag{4}$$

$$
\begin{prooftree}
\AxiomC{$[\psi]^a$}
\AxiomC{$[\neg\psi]^b$}
\RightLabel{$(\neg_e)$}
\BinaryInfC{$\bot$}
\RightLabel{(PBC)$^b$}
\UnaryInfC{$\psi$}
\AxiomC{$[\exists_m \psi]^c$}
\RightLabel{$(\exists_e)^a$}
\BinaryInfC{$\psi$}
\RightLabel{$(\forall_i)$}
\UnaryInfC{$\forall_m \psi$}
\AxiomC{$\forall_m \psi \to \forall_n \psi$}
\RightLabel{$(\to_e)$}
\BinaryInfC{$\forall_n \psi$}
\RightLabel{$(\to_i)^c$}
\UnaryInfC{$\exists_m \psi \to \forall_n \psi$}
\end{prooftree}
$$

Apesar de muito elegante essa prova ela está incompleta e possivelmente errada, pois não podemos fazer o passo $\exists_e$ do jeito que foi feito, por termos uma hipótese com variável livre como resultado, o que nos impede de usar esta regra nesse passo.

### 4.2 Problema 2

**Problema 2**: Agora utilizaremos o conhecimento adquirido sobre indução para provar a correção de um algoritmo de ordenação de listas conhecido como “insertion sort”, ou ordenação por inserção. O pseudocódigo deste algoritmo é dado a seguir:

$$
\text{insertionSort}(l) = \begin{cases} l, & \text{se } l = [\,] \\ \text{insert}(h, \text{insertSort}(l')), & \text{se } l = h :: l' \end{cases}
$$

onde

$$
\text{insert}(x, l) = \begin{cases} x :: [\,], & \text{se } l = [\,] \\ x :: l, & \text{se } l = h :: l' \text{ e } x \leq h \\ , & \text{ e } x > h \end{cases}
$$

Nosso objetivo é provar que o algoritmo acima é correto, mas o que isto significa? Como expressar este fato por meio de um teorema? Quais passos intermediários você julga que serão necessários para provar este teorema? Explique e justifique sua solução, e os passos que utilizou para obtê-la, de forma clara e completa por meio de um relatório detalhado[1].

## 5 Conclusão

## Referências


[1] Flávio L. C. de Moura (Prof.). *Indução Matemática*.