Demonstrações por Indução

2020/2021

Princípio de Indução Matemática - intuição

Suponhamos que existe uma linha infinita de pessoas, numeradas $1, 2, 3, \ldots$ e que cada pessoa foi instruída do seguinte: “Se alguma coisa lhe for segredada ao ouvido, segrede o mesmo à pessoa à sua frente (cujo número é maior)”. O que acontece se for algo for segredado à pessoa 1? 1 vai contar a 2, 2 vai contar a 3, 3 vai contar a 4, e assim sucessivamente todas as pessoas vão conhecer o segredo!

Princípio de Indução Matemática - intuição

Suponhamos que existe uma linha infinita de pessoas, numeradas $1, 2, 3, \ldots$ e que cada pessoa foi instruída do seguinte: “Se alguma coisa lhe for segredada ao ouvido, segrede o mesmo à pessoa à sua frente (cujo número é maior)”. O que acontece se for algo for segredado à pessoa 1? 1 vai contar a 2, 2 vai contar a 3, 3 vai contar a 4, e assim sucessivamente todas as pessoas vão conhecer o segredo!

De igual forma, suponhamos que alinhamos um número infinito de peças de dominó, tais que se qualquer peça de dominó cair, então a seguinte também cai. O que acontece se derrubarmos a primeira peça? As peças caem todas.

Princípio de Indução Matemática

Sejam $n_0 \in \mathbb{Z}$ e $S = \{\ n \in \mathbb{Z} \mid n \geq n_0\ \}$, e suponhamos que $P(n)$ abrevia a condição "o inteiro n satisfaz a condição P".

Princípio de Indução Matemática

Sejam $n_0 \in \mathbb{Z}$ e $S = \{\, n \in \mathbb{Z} \mid n \geq n_0 \,\}$, e suponhamos que $P(n)$ abrevia a condição "o inteiro n satisfaz a condição P".
Se se verificarem as condições seguintes,

- $P(n_0)$ (i.e. a condição P verifica-se para n_0);

Princípio de Indução Matemática

Sejam $n_0 \in \mathbb{Z}$ e $S = \{\, n \in \mathbb{Z} \mid n \geq n_0 \,\}$, e suponhamos que $P(n)$ abrevia a condição "o inteiro n satisfaz a condição P".
Se se verificarem as condições seguintes,

- $P(n_0)$ (i.e. a condição P verifica-se para n_0);
- para todo o inteiro $n \geq n_0$, $P(n)$ implica $P(n+1)$;

Princípio de Indução Matemática

Sejam $n_0 \in \mathbb{Z}$ e $S = \{\, n \in \mathbb{Z} \mid n \geq n_0 \,\}$, e suponhamos que $P(n)$ abrevia a condição "o inteiro n satisfaz a condição P". Se se verificarem as condições seguintes,

- $P(n_0)$ (i.e. a condição P verifica-se para n_0);
- para todo o inteiro $n \geq n_0$, $P(n)$ implica $P(n+1)$;

então temos $P(n)$ para todo $n \geq n_0$, i.e. para todo $n \in S$.

Princípio de Indução Matemática

Sejam $n_0 \in \mathbb{Z}$ e $S = \{\, n \in \mathbb{Z} \mid n \geq n_0 \,\}$, e suponhamos que $P(n)$ abrevia a condição "o inteiro n satisfaz a condição P". Se se verificarem as condições seguintes,

- $P(n_0)$ (i.e. a condição P verifica-se para n_0);
- para todo o inteiro $n \geq n_0$, $P(n)$ implica $P(n+1)$;

então temos $P(n)$ para todo $n \geq n_0$, i.e. para todo $n \in S$.

CB: À condição $P(n_0)$ chamamos geralmente **caso base**,

Princípio de Indução Matemática

Sejam $n_0 \in \mathbb{Z}$ e $S = \{\, n \in \mathbb{Z} \mid n \geq n_0 \,\}$, e suponhamos que $P(n)$ abrevia a condição "o inteiro n satisfaz a condição P". Se se verificarem as condições seguintes,

- $P(n_0)$ (i.e. a condição P verifica-se para n_0);
- para todo o inteiro $n \geq n_0$, $P(n)$ implica $P(n+1)$;

então temos $P(n)$ para todo $n \geq n_0$, i.e. para todo $n \in S$.

CB: À condição $P(n_0)$ chamamos geralmente **caso base**,

H: e a $\forall n \geq n_0\ [P(n) \Rightarrow P(n+1)]$ **condição de hereditariedade** ou **passo de indução**.

Indução Matemática

- Começamos por provar que o predicado é válido para o menor elemento de S (0 se $S = \mathbb{N}$).

Indução Matemática

- Começamos por provar que o predicado é válido para o menor elemento de S (0 se $S = \mathbb{N}$).
- Depois provamos que se o predicado é válido para um elemento n então também é válido para o próximo elemento no conjunto, i.e. $n + 1$.

Indução Matemática

- Começamos por provar que o predicado é válido para o menor elemento de S (0 se $S = \mathbb{N}$).
- Depois provamos que se o predicado é válido para um elemento n então também é válido para o próximo elemento no conjunto, i.e. $n + 1$.

Logo:

- Como é válido para o primeiro elemento então é válido para o segundo.

Indução Matemática

- Começamos por provar que o predicado é válido para o menor elemento de S (0 se $S = \mathbb{N}$).
- Depois provamos que se o predicado é válido para um elemento n então também é válido para o próximo elemento no conjunto, i.e. $n + 1$.

Logo:

- Como é válido para o primeiro elemento então é válido para o segundo.
- Como é válido para o segundo elemento então é válido para o terceiro.

Indução Matemática

- Começamos por provar que o predicado é válido para o menor elemento de S (0 se $S = \mathbb{N}$).
- Depois provamos que se o predicado é válido para um elemento n então também é válido para o próximo elemento no conjunto, i.e. $n + 1$.

Logo:

- Como é válido para o primeiro elemento então é válido para o segundo.
- Como é válido para o segundo elemento então é válido para o terceiro.
- Como é válido para o terceiro elemento então é válido para o quarto.
- ...

Indução Matemática

Supondo que queremos demonstrar por indução que uma propriedade P se verifica para todo $n \in S = \{ n \in \mathbb{Z} \mid n \geq n_0 \}$. Procedemos da seguinte forma:

Indução Matemática

Supondo que queremos demonstrar por indução que uma propriedade P se verifica para todo $n \in S = \{ n \in \mathbb{Z} \mid n \geq n_0 \}$. Procedemos da seguinte forma:

1. Enunciamos o método de prova. Por exemplo, "Prosseguimos por indução."

Indução Matemática

Supondo que queremos demonstrar por indução que uma propriedade P se verifica para todo $n \in S = \{ n \in \mathbb{Z} \mid n \geq n_0 \}$. Procedemos da seguinte forma:

1. Enunciamos o método de prova. Por exemplo, "Prosseguimos por indução."
2. Demonstramos a "base de indução": $P(n_0)$. Ou seja, que P se verifica para o primeiro elemento do conjunto S.

Indução Matemática

Supondo que queremos demonstrar por indução que uma propriedade P se verifica para todo $n \in S = \{ n \in \mathbb{Z} \mid n \geq n_0 \}$. Procedemos da seguinte forma:

1. Enunciamos o método de prova. Por exemplo, "Prosseguimos por indução."
2. Demonstramos a "base de indução": $P(n_0)$. Ou seja, que P se verifica para o primeiro elemento do conjunto S.
3. Assumimos a "hipótese de indução": $P(n)$ para um qualquer inteiro $n \geq n_0$. Ou seja, assumimos que a propriedade se verifica para um inteiro $n \geq n_0$.

Indução Matemática

Supondo que queremos demonstrar por indução que uma propriedade P se verifica para todo $n \in S = \{ n \in \mathbb{Z} \mid n \geq n_0 \}$. Procedemos da seguinte forma:

1. Enunciamos o método de prova. Por exemplo, "Prosseguimos por indução."
2. Demonstramos a "base de indução": $P(n_0)$. Ou seja, que P se verifica para o primeiro elemento do conjunto S.
3. Assumimos a "hipótese de indução": $P(n)$ para um qualquer inteiro $n \geq n_0$. Ou seja, assumimos que a propriedade se verifica para um inteiro $n \geq n_0$.
4. Demonstramos, usando a hipótese de indução, que a propriedade P se verifica para $n+1$, ou seja, $P(n+1)$.

Indução Matemática

Supondo que queremos demonstrar por indução que uma propriedade P se verifica para todo $n \in S = \{ n \in \mathbb{Z} \mid n \geq n_0 \}$. Procedemos da seguinte forma:

1. Enunciamos o método de prova. Por exemplo, "Prosseguimos por indução."
2. Demonstramos a "base de indução": $P(n_0)$. Ou seja, que P se verifica para o primeiro elemento do conjunto S.
3. Assumimos a "hipótese de indução": $P(n)$ para um qualquer inteiro $n \geq n_0$. Ou seja, assumimos que a propriedade se verifica para um inteiro $n \geq n_0$.
4. Demonstramos, usando a hipótese de indução, que a propriedade P se verifica para $n+1$, ou seja, $P(n+1)$.
5. Concluir a prova.

Indução Matemática

Supondo que queremos demonstrar por indução que uma propriedade P se verifica para todo $n \in S = \{ n \in \mathbb{Z} \mid n \geq n_0 \}$. Procedemos da seguinte forma:

1. Enunciamos o método de prova. Por exemplo, "Prosseguimos por indução."
2. Demonstramos a "base de indução": $P(n_0)$. Ou seja, que P se verifica para o primeiro elemento do conjunto S.
3. Assumimos a "hipótese de indução": $P(n)$ para um qualquer inteiro $n \geq n_0$. Ou seja, assumimos que a propriedade se verifica para um inteiro $n \geq n_0$.
4. Demonstramos, usando a hipótese de indução, que a propriedade P se verifica para $n+1$, ou seja, $P(n+1)$.
5. Concluir a prova.

Indução Matemática: exemplo

Vamos mostrar que

$$\forall n \in \mathbb{N} \quad \sum_{i=0}^{n} i = \frac{n(n+1)}{2}.$$

Indução Matemática: exemplo

Vamos mostrar que

$$\forall n \in \mathbb{N} \quad \sum_{i=0}^{n} i = \frac{n(n+1)}{2}.$$

- Começamos por provar o caso base, i.e. $\sum_{i=0}^{0} i = \frac{0(0+1)}{2}$

Indução Matemática: exemplo

Vamos mostrar que

$$\forall n \in \mathbb{N} \quad \sum_{i=0}^{n} i = \frac{n(n+1)}{2}.$$

- Começamos por provar o caso base, i.e. $\sum_{i=0}^{0} i = \frac{0(0+1)}{2}$
- De seguida tomamos um inteiro $n \geq 0$ e suponhamos $P(n)$:

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2}.$$

- Usando um método de prova directa (veremos mais à frente) provamos $P(n+1)$:

$$\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2},$$

podendo para isso usar a hipótese de indução $P(n)$.

Indução Matemática: exemplo (prova completa)

Vamos demonstrar por indução que, para todo o número natural n: $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

Indução Matemática: exemplo (prova completa)

Vamos demonstrar por indução que, para todo o número natural n: $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

- Caso base: Para $n = 0$ temos $\sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$.

Indução Matemática: exemplo (prova completa)

Vamos demonstrar por indução que, para todo o número natural n: $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

- Caso base: Para $n = 0$ temos $\sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$.
- Passo de indução: Seja $n \geq 0$ e suponhamos que $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$. Queremos mostrar $\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$.

Indução Matemática: exemplo (prova completa)

Vamos demonstrar por indução que, para todo o número natural n: $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

- Caso base: Para $n = 0$ temos $\sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$.
- Passo de indução: Seja $n \geq 0$ e suponhamos que $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$. Queremos mostrar $\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$.

$$\sum_{i=0}^{n+1} i \quad = \quad \sum_{i=0}^{n} i + \sum_{i=n+1}^{n+1} i \quad \text{(propriedade dos somatórios)}$$

Indução Matemática: exemplo (prova completa)

Vamos demonstrar por indução que, para todo o número natural n: $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

- Caso base: Para $n = 0$ temos $\sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$.
- Passo de indução: Seja $n \geq 0$ e suponhamos que $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$. Queremos mostrar $\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$.

$$\begin{aligned} \sum_{i=0}^{n+1} i &= \sum_{i=0}^{n} i + \sum_{i=n+1}^{n+1} i \quad \text{(propriedade dos somatórios)} \\ &= \sum_{i=0}^{n} i + n + 1 \end{aligned}$$

Indução Matemática: exemplo (prova completa)

Vamos demonstrar por indução que, para todo o número natural n: $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

- Caso base: Para $n = 0$ temos $\sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$.
- Passo de indução: Seja $n \geq 0$ e suponhamos que $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$. Queremos mostrar $\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$.

$$
\begin{array}{rcll}
\sum_{i=0}^{n+1} i & = & \sum_{i=0}^{n} i + \sum_{i=n+1}^{n+1} i & \text{(propriedade dos somatórios)} \\
& = & \sum_{i=0}^{n} i + n + 1 & \\
& = & \frac{n(n+1)}{2} + (n+1) & \text{(usando a hipótese de indução)}
\end{array}
$$

Indução Matemática: exemplo (prova completa)

Vamos demonstrar por indução que, para todo o número natural n: $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

- Caso base: Para $n = 0$ temos $\sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$.
- Passo de indução: Seja $n \geq 0$ e suponhamos que $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$. Queremos mostrar $\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$.

$$
\begin{array}{rcll}
\sum_{i=0}^{n+1} i & = & \sum_{i=0}^{n} i + \sum_{i=n+1}^{n+1} i & \text{(propriedade dos somatórios)} \\
& = & \sum_{i=0}^{n} i + n + 1 & \\
& = & \frac{n(n+1)}{2} + (n+1) & \text{(usando a hipótese de indução)} \\
& = & \frac{n(n+1)+2(n+1)}{2} & \text{(por manipulação algébrica)}
\end{array}
$$

Indução Matemática: exemplo (prova completa)

Vamos demonstrar por indução que, para todo o número natural n: $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

- Caso base: Para $n = 0$ temos $\sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$.
- Passo de indução: Seja $n \geq 0$ e suponhamos que $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$. Queremos mostrar $\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$.

$$
\begin{array}{rcll}
\sum_{i=0}^{n+1} i & = & \sum_{i=0}^{n} i + \sum_{i=n+1}^{n+1} i & \text{(propriedade dos somatórios)} \\
& = & \sum_{i=0}^{n} i + n + 1 & \\
& = & \frac{n(n+1)}{2} + (n+1) & \text{(usando a hipótese de indução)} \\
& = & \frac{n(n+1)+2(n+1)}{2} & \text{(por manipulação algébrica)} \\
& = & \frac{(n+1)((n+1)+1)}{2} &
\end{array}
$$

Indução Matemática: exemplo (prova completa)

Vamos demonstrar por indução que, para todo o número natural n: $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

- Caso base: Para $n = 0$ temos $\sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$.
- Passo de indução: Seja $n \geq 0$ e suponhamos que $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$. Queremos mostrar $\sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2}$.

$$
\begin{array}{rcll}
\sum_{i=0}^{n+1} i & = & \sum_{i=0}^{n} i + \sum_{i=n+1}^{n+1} i & \text{(propriedade dos somatórios)} \\
& = & \sum_{i=0}^{n} i + n + 1 & \\
& = & \frac{n(n+1)}{2} + (n+1) & \text{(usando a hipótese de indução)} \\
& = & \frac{n(n+1)+2(n+1)}{2} & \text{(por manipulação algébrica)} \\
& = & \frac{(n+1)((n+1)+1)}{2} &
\end{array}
$$

Logo, para qualquer $n \in \mathbb{N}$, $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$.

Indução Matemática: exemplo II

Teorema: Qualquer potência inteira positiva de 3 é ímpar.

Indução Matemática: exemplo II

Teorema: Qualquer potência inteira positiva de 3 é ímpar.
Prova: Queremos mostrar que para qualquer $n \geq 1$ se tem $3^n = 2l + 1$, para algum $l \in \mathbb{N}$. Prosseguimos por indução.

Indução Matemática: exemplo II

Teorema: Qualquer potência inteira positiva de 3 é ímpar.
Prova: Queremos mostrar que para qualquer $n \geq 1$ se tem $3^n = 2l + 1$, para algum $l \in \mathbb{N}$. Prosseguimos por indução.

- Caso base: para $n = 1$ temos $3 = 2 \cdot 1 + 1$, o que verifica a propriedade.

Indução Matemática: exemplo II

Teorema: Qualquer potência inteira positiva de 3 é ímpar.
Prova: Queremos mostrar que para qualquer $n \geq 1$ se tem $3^n = 2l + 1$, para algum $l \in \mathbb{N}$. Prosseguimos por indução.

- Caso base: para $n = 1$ temos $3 = 2 \cdot 1 + 1$, o que verifica a propriedade.
- Passo de indução: Seja $n \geq 1$ e suponhamos que 3^n é ímpar (ou seja $3^n = 2l + 1$, para algum $l \in \mathbb{N}$). Queremos demonstrar que 3^{n+1} também é ímpar:

Indução Matemática: exemplo II

Teorema: Qualquer potência inteira positiva de 3 é ímpar.
Prova: Queremos mostrar que para qualquer $n \geq 1$ se tem $3^n = 2l + 1$, para algum $l \in \mathbb{N}$. Prosseguimos por indução.

- Caso base: para $n = 1$ temos $3 = 2 \cdot 1 + 1$, o que verifica a propriedade.
- Passo de indução: Seja $n \geq 1$ e suponhamos que 3^n é ímpar (ou seja $3^n = 2l + 1$, para algum $l \in \mathbb{N}$). Queremos demonstrar que 3^{n+1} também é ímpar:

$$3^{n+1} = 3 \cdot 3^n$$

Indução Matemática: exemplo II

Teorema: Qualquer potência inteira positiva de 3 é ímpar.
Prova: Queremos mostrar que para qualquer $n \geq 1$ se tem $3^n = 2l + 1$, para algum $l \in \mathbb{N}$. Prosseguimos por indução.

- Caso base: para $n = 1$ temos $3 = 2 \cdot 1 + 1$, o que verifica a propriedade.
- Passo de indução: Seja $n \geq 1$ e suponhamos que 3^n é ímpar (ou seja $3^n = 2l + 1$, para algum $l \in \mathbb{N}$). Queremos demonstrar que 3^{n+1} também é ímpar:

$$\begin{aligned} 3^{n+1} &= 3 \cdot 3^n \\ &= 3(2l + 1) \qquad \text{(hipótese de indução)} \end{aligned}$$

Indução Matemática: exemplo II

Teorema: Qualquer potência inteira positiva de 3 é ímpar.
Prova: Queremos mostrar que para qualquer $n \geq 1$ se tem $3^n = 2l + 1$, para algum $l \in \mathbb{N}$. Prosseguimos por indução.

- Caso base: para $n = 1$ temos $3 = 2 \cdot 1 + 1$, o que verifica a propriedade.
- Passo de indução: Seja $n \geq 1$ e suponhamos que 3^n é ímpar (ou seja $3^n = 2l + 1$, para algum $l \in \mathbb{N}$). Queremos demonstrar que 3^{n+1} também é ímpar:

$$\begin{aligned} 3^{n+1} &= 3 \cdot 3^n \\ &= 3(2l + 1) \qquad \text{(hipótese de indução)} \\ &= 2(3l + 1) + 1. \end{aligned}$$

Indução Matemática: exemplo II

Teorema: Qualquer potência inteira positiva de 3 é ímpar.
Prova: Queremos mostrar que para qualquer $n \geq 1$ se tem $3^n = 2l + 1$, para algum $l \in \mathbb{N}$. Prosseguimos por indução.

- Caso base: para $n = 1$ temos $3 = 2 \cdot 1 + 1$, o que verifica a propriedade.
- Passo de indução: Seja $n \geq 1$ e suponhamos que 3^n é ímpar (ou seja $3^n = 2l + 1$, para algum $l \in \mathbb{N}$). Queremos demonstrar que 3^{n+1} também é ímpar:

$$\begin{aligned} 3^{n+1} &= 3 \cdot 3^n \\ &= 3(2l + 1) \qquad \text{(hipótese de indução)} \\ &= 2(3l + 1) + 1. \end{aligned}$$

Como $3l + 1 \in \mathbb{N}$ concluimos que 3^{n+1} é ímpar, o que termina a demonstração.

Exercícios:

- Mostre que $\sum_{i=0}^{n} r^i = \frac{r^{n+1}-1}{r-1}$, para todo o inteiro $n \geq 0$ e todo o r real excepto 1.
- Mostre que para todo o inteiro $n \geq 1$, $2^{2n} - 1$ é divisível por 3.
- Mostre que para todo o inteiro $n \geq 3$, $2n + 1 \leq 2^n$.
- Considere a sucessão $a_1, a_2, a_3, \ldots$ definida por:

$$\begin{aligned} a_1 &= 1 \\ a_2 &= 3 \\ a_n &= a_{n-2} + 2a_{n-1}, n \geq 3 \end{aligned}$$

  - Determine os valores de a_3, a_4, a_5, a_6 e a_7.
  - Mostre que $\forall n \geq 1$, a_n é um inteiro positivo ímpar.

Indução Matemática Forte

A regra de inferência é ligeiramente modificada:

Indução Matemática Forte

A regra de inferência é ligeiramente modificada:

- Se $P(n_0)$ se verifica, e

Indução Matemática Forte

A regra de inferência é ligeiramente modificada:

- Se $P(n_0)$ se verifica, e
- Se $P(n_0), P(n_0+1), \ldots, P(n)$ se verificam (para um n arbitrário), implica que $P(n+1)$ se verifica

Indução Matemática Forte

A regra de inferência é ligeiramente modificada:

- Se $P(n_0)$ se verifica, e
- Se $P(n_0), P(n_0+1), \ldots, P(n)$ se verificam (para um n arbitrário), implica que $P(n+1)$ se verifica

então $\forall n \geq n_0\ P(n)$.

Indução Matemática Forte

A regra de inferência é ligeiramente modificada:

- Se $P(n_0)$ se verifica, e
- Se $P(n_0), P(n_0+1), \ldots, P(n)$ se verificam (para um n arbitrário), implica que $P(n+1)$ se verifica

então $\forall n \geq n_0 \; P(n)$.

As hipóteses são:

CB: $P(n_0)$

H: $\forall n \geq n_0 \; [(\forall k, n_0 \leq k \leq n, P(k)) \; \Rightarrow P(n+1)]$.

Indução Matemática Forte

A regra de inferência é ligeiramente modificada:

- Se $P(n_0)$ se verifica, e
- Se $P(n_0), P(n_0+1), \ldots, P(n)$ se verificam (para um n arbitrário), implica que $P(n+1)$ se verifica

então $\forall n \geq n_0\ P(n)$.

As hipóteses são:

CB: $P(n_0)$

H: $\forall n \geq n_0\ [(\forall k, n_0 \leq k \leq n, P(k)) \Rightarrow P(n+1)]$.

As duas regras são equivalentes, no entanto algumas vezes a indução forte é de mais fácil aplicação.

Indução Forte: exemplo

Teorema: Qualquer inteiro positivo maior do que 1 pode ser escrito como um produto de primos.

Indução Forte: exemplo

Teorema: Qualquer inteiro positivo maior do que 1 pode ser escrito como um produto de primos.
Prova: A prova prossegue por indução forte:

- Caso base, $n = 2$: como 2 é um número primo, a propriedade verifica-se.

Indução Forte: exemplo

Teorema: Qualquer inteiro positivo maior do que 1 pode ser escrito como um produto de primos.
Prova: A prova prossegue por indução forte:

- Caso base, $n = 2$: como 2 é um número primo, a propriedade verifica-se.
- Passo de indução: Vamos assumir que a propriedade se verifica para todos os inteiros entre 2 e n.

Indução Forte: exemplo

Teorema: Qualquer inteiro positivo maior do que 1 pode ser escrito como um produto de primos.
Prova: A prova prossegue por indução forte:

- Caso base, $n = 2$: como 2 é um número primo, a propriedade verifica-se.
- Passo de indução: Vamos assumir que a propriedade se verifica para todos os inteiros entre 2 e n. Se $n + 1$ é um número primo então a propriedade verifica-se trivialmente.

Indução Forte: exemplo

Teorema: Qualquer inteiro positivo maior do que 1 pode ser escrito como um produto de primos.
Prova: A prova prossegue por indução forte:

- Caso base, $n = 2$: como 2 é um número primo, a propriedade verifica-se.
- Passo de indução: Vamos assumir que a propriedade se verifica para todos os inteiros entre 2 e n. Se $n + 1$ é um número primo então a propriedade verifica-se trivialmente. Caso contrário, $n + 1$ tem um divisor diferente de 1 e $n + 1$.

Indução Forte: exemplo

Teorema: Qualquer inteiro positivo maior do que 1 pode ser escrito como um produto de primos.
Prova: A prova prossegue por indução forte:

- Caso base, $n = 2$: como 2 é um número primo, a propriedade verifica-se.
- Passo de indução: Vamos assumir que a propriedade se verifica para todos os inteiros entre 2 e n. Se $n + 1$ é um número primo então a propriedade verifica-se trivialmente. Caso contrário, $n + 1$ tem um divisor diferente de 1 e $n + 1$. Logo $n + 1 = a \cdot b$, com $2 \leq a, b \leq n$.

Indução Forte: exemplo

Teorema: Qualquer inteiro positivo maior do que 1 pode ser escrito como um produto de primos.
Prova: A prova prossegue por indução forte:

- Caso base, $n = 2$: como 2 é um número primo, a propriedade verifica-se.
- Passo de indução: Vamos assumir que a propriedade se verifica para todos os inteiros entre 2 e n. Se $n + 1$ é um número primo então a propriedade verifica-se trivialmente. Caso contrário, $n + 1$ tem um divisor diferente de 1 e $n + 1$. Logo $n + 1 = a \cdot b$, com $2 \leq a, b \leq n$. Por hipótese de indução, ambos a e b podem ser escritos como um produto de primos,

Indução Forte: exemplo

Teorema: Qualquer inteiro positivo maior do que 1 pode ser escrito como um produto de primos.
Prova: A prova prossegue por indução forte:

- Caso base, $n = 2$: como 2 é um número primo, a propriedade verifica-se.
- Passo de indução: Vamos assumir que a propriedade se verifica para todos os inteiros entre 2 e n. Se $n + 1$ é um número primo então a propriedade verifica-se trivialmente. Caso contrário, $n + 1$ tem um divisor diferente de 1 e $n + 1$. Logo $n + 1 = a \cdot b$, com $2 \leq a, b \leq n$. Por hipótese de indução, ambos a e b podem ser escritos como um produto de primos, logo também o seu produto pode ser expresso como um produto de primos.

Indução Forte: exemplo

Teorema: Qualquer inteiro positivo maior do que 1 pode ser escrito como um produto de primos.
Prova: A prova prossegue por indução forte:

- Caso base, $n = 2$: como 2 é um número primo, a propriedade verifica-se.
- Passo de indução: Vamos assumir que a propriedade se verifica para todos os inteiros entre 2 e n. Se $n + 1$ é um número primo então a propriedade verifica-se trivialmente. Caso contrário, $n + 1$ tem um divisor diferente de 1 e $n + 1$. Logo $n + 1 = a \cdot b$, com $2 \leq a, b \leq n$. Por hipótese de indução, ambos a e b podem ser escritos como um produto de primos, logo também o seu produto pode ser expresso como um produto de primos.

Logo, verifica-se por indução, que qualquer inteiro positivo maior do que 1 pode ser escrito como um produto de primos.

Indução forte: exemplo

O jogo das cartas: Considere um jogo em que dois jogadores removem alternadamente um número arbitrário, mas não nulo, de cartas de uma de duas pilhas (de cartas) inicialmente não vazias. O jogador que remover a ultima carta ganha o jogo. Mostre que, se as duas pilhas contiverem inicialmente o mesmo numero de cartas, então o segundo jogador tem uma estratégia que lhe garante a vitoria do jogo.

Exercício

Considere o conjunto $\mathcal{F}$ de fórmulas do cálculo proposicional definidas pelas regras seguintes:

- qualquer variável proposicional *p* é uma fórmula em $\mathcal{F}$;
- se $\alpha, \beta \in \mathcal{F}$, então $(\neg\alpha) \in \mathcal{F}$ e $(\alpha \rightarrow \beta) \in \mathcal{F}$.

Exercício

Considere o conjunto $\mathcal{F}$ de fórmulas do cálculo proposicional definidas pelas regras seguintes:

- qualquer variável proposicional *p* é uma fórmula em $\mathcal{F}$;
- se $\alpha, \beta \in \mathcal{F}$, então $(\neg\alpha) \in \mathcal{F}$ e $(\alpha \to \beta) \in \mathcal{F}$.

Denotamos por $|\alpha|$, por $|\alpha|_{\to}$ e por $|\alpha|_{var}$ respectivamente o número de símbolos em α (váriaveis proposicionais, implicações, negações e parênteses), o número de implicações em α e o número de variáveis proposicionais em α.

Exercício

Considere o conjunto $\mathcal{F}$ de fórmulas do cálculo proposicional definidas pelas regras seguintes:

- qualquer variável proposicional *p* é uma fórmula em $\mathcal{F}$;
- se $\alpha, \beta \in \mathcal{F}$, então $(\neg\alpha) \in \mathcal{F}$ e $(\alpha \to \beta) \in \mathcal{F}$.

Denotamos por $|\alpha|$, por $|\alpha|_{\to}$ e por $|\alpha|_{var}$ respectivamente o número de símbolos em α (váriaveis proposicionais, implicações, negações e parênteses), o número de implicações em α e o número de variáveis proposicionais em α.

Mostre por indução sobre $n = |\alpha|$ que para qualquer fórmula α de comprimento $n = |\alpha| \geq 1$ se tem $|\alpha|_{var} = |\alpha|_{\to} + 1$.

Raciocínio equacional

Para simplificar expressões matemáticas podemos usar igualdades algébricas como regras de reescrita.

Este tipo manipulação chama-se raciocínio equacional.

Algumas igualdades algébricas

$$x + y = y + x \qquad \text{comutatividade de } +$$

$$x * y = y * x \qquad \text{comutatividade de } *$$

$$x + (y + z) = (x + y) + z \qquad \text{associatividade de } +$$

$$x * (y * z) = (x * y) * z \qquad \text{associatividade de } *$$

$$x * (y + z) = x * y + x * z \qquad \text{distributividade de } * \text{ sobre } +$$

Podemos substituir os lados esquerdos pelos lados direitos ou vice-versa.

Exemplo

$$(x + y) * (x + y)$$

Exemplo

$$
\begin{aligned}
& (x + y) * (x + y) && \{\text{distributividade}\} \\
= \quad & (x + y) * x + (x + y) * y
\end{aligned}
$$

Exemplo

$$
\begin{aligned}
& (x + y) * (x + y) && \{\text{distributividade}\} \\
= \quad & (x + y) * x + (x + y) * y && \{\text{comutatividade de } *\} \\
= \quad & x * (x + y) + (x + y) * y &&
\end{aligned}
$$

Exemplo

$$
\begin{aligned}
& (x + y) * (x + y) && \{\text{distributividade}\} \\
= \quad & (x + y) * x + (x + y) * y && \{\text{comutatividade de *}\} \\
= \quad & x * (x + y) + (x + y) * y && \{\text{distributividade}\} \\
= \quad & x * x + x * y + (x + y) * y
\end{aligned}
$$

Exemplo

$$
\begin{aligned}
& (x+y)*(x+y) && \{\text{distributividade}\} \\
= \; & (x+y)*x+(x+y)*y && \{\text{comutatividade de } *\} \\
= \; & x*(x+y)+(x+y)*y && \{\text{distributividade}\} \\
= \; & x*x+x*y+(x+y)*y && \{\text{comutatividade de } *\} \\
= \; & x*x+x*y+y*(x+y)
\end{aligned}
$$

Exemplo

$$
\begin{aligned}
& (x + y) * (x + y) && \{\text{distributividade}\} \\
= \; & (x + y) * x + (x + y) * y && \{\text{comutatividade de } *\} \\
= \; & x * (x + y) + (x + y) * y && \{\text{distributividade}\} \\
= \; & x * x + x * y + (x + y) * y && \{\text{comutatividade de } *\} \\
= \; & x * x + x * y + y * (x + y) && \{\text{distributividade}\} \\
= \; & x * x + x * y + y * x + y * y
\end{aligned}
$$

Exemplo

$$
\begin{aligned}
& (x+y)*(x+y) && \{\text{distributividade}\} \\
= \; & (x+y)*x+(x+y)*y && \{\text{comutatividade de *}\} \\
= \; & x*(x+y)+(x+y)*y && \{\text{distributividade}\} \\
= \; & x*x+x*y+(x+y)*y && \{\text{comutatividade de *}\} \\
= \; & x*x+x*y+y*(x+y) && \{\text{distributividade}\} \\
= \; & x*x+x*y+y*x+y*y && \{\text{comutatividade de *}\} \\
= \; & x*x+x*y+x*y+y*y
\end{aligned}
$$

Exemplo

$$
\begin{aligned}
& (x + y) * (x + y) && \{\text{distributividade}\} \\
= \; & (x + y) * x + (x + y) * y && \{\text{comutatividade de } *\} \\
= \; & x * (x + y) + (x + y) * y && \{\text{distributividade}\} \\
= \; & x * x + x * y + (x + y) * y && \{\text{comutatividade de } *\} \\
= \; & x * x + x * y + y * (x + y) && \{\text{distributividade}\} \\
= \; & x * x + x * y + y * x + y * y && \{\text{comutatividade de } *\} \\
= \; & x * x + x * y + x * y + y * y && \{\text{distributividade}\} \\
= \; & x * x + (1 + 1) * x * y + y * y
\end{aligned}
$$

Exemplo

$$
\begin{aligned}
& (x+y)*(x+y) && \{\text{distributividade}\} \\
= \; & (x+y)*x+(x+y)*y && \{\text{comutatividade de } *\} \\
= \; & x*(x+y)+(x+y)*y && \{\text{distributividade}\} \\
= \; & x*x+x*y+(x+y)*y && \{\text{comutatividade de } *\} \\
= \; & x*x+x*y+y*(x+y) && \{\text{distributividade}\} \\
= \; & x*x+x*y+y*x+y*y && \{\text{comutatividade de } *\} \\
= \; & x*x+x*y+x*y+y*y && \{\text{distributividade}\} \\
= \; & x*x+(1+1)*x*y+y*y && \{\text{abreviaturas}\} \\
= \; & x^2+2xy+y^2
\end{aligned}
$$

Raciocínio equacional sobre programas I

Podemos mostrar propriedades de programas usando definições de funções como regras de re-escrita.

Raciocínio equacional sobre programas II

Vamos mostrar que

```
reverse [x] = [x]
```

usando as definições seguintes:

```
reverse []     = []                    (reverse.1)
reverse (x:xs) = reverse xs ++ [x]     (reverse.2)

[]     ++ ys = ys                      (++.1)
(x:xs) ++ ys = x:(xs++ys)              (++.2)
```

Exemplo

Começamos pelo lado esquerdo:

```
reverse [x]
```

Exemplo

Começamos pelo lado esquerdo:

```
    reverse [x]
```

{notação de listas}

```
  = reverse (x:[])
```

Exemplo

Começamos pelo lado esquerdo:

```
    reverse [x]
```

{notação de listas}

```
  = reverse (x:[])
```

{reverse.2}

```
  = reverse [] ++ [x]
```

Exemplo

Começamos pelo lado esquerdo:

```
    reverse [x]
```
{notação de listas}
```
  = reverse (x:[])
```
{reverse.2}
```
  = reverse [] ++ [x]
```
{reverse.1}
```
  = [] ++ [x]
```

Exemplo

Começamos pelo lado esquerdo:

```
    reverse [x]
```
{notação de listas}
```
  = reverse (x:[])
```
{reverse.2}
```
  = reverse [] ++ [x]
```
{reverse.1}
```
  = [] ++ [x]
```
{++.1}
```
  = [x]
```

Obtemos a expressão do lado direito.

Porquê provar propriedades de programas?

- Verificação formal da correcção
  1. provar propriedades universais
  2. garantia de resultados correctos para *quaisquer* valores
  3. garantia de terminação e ausência de erros
- Simplificação e transformação
  1. transformar programas usando igualdades
  2. sintetizar programas apartir de requisitos (especificações)
  3. obter um programa eficiente a partir de um mais simples

"Testing shows the presence, not the absence of bugs."

— E. Disjkstra

Porquê em Haskell?

Podemos usar raciocínio equacional sobre programas Haskell porque são definidos por *equações*.

Por contraposição: programas imperativos são definidos por *sequências de instruções* – não são equações.

Exemplo

Após a instrução

```
n = n+1;          // em C,C++,Java...
```

não podemos substituir n por $n + 1$ — trata-se duma atribuição e não duma equação.

Recursão e indução

Em programação usamos recursão para definir funções sobre números naturais, listas, etc.

Além de raciocínio equacional, necessitamos de indução matemática para provar propriedades dessas funções.

Exemplo

```
length :: [a] -> Int
length []     = 0                          (length.1)
length (x:xs) = 1 + length xs              (length.2)

replicate :: Int -> a -> [a]
replicate 0 x  = []                        (replicate.1)
replicate n x | n>0
    = x : replicate (n-1) x                (replicate.2)
```

Vamos mostrar

```
length (replicate n x) = n
```

usando indução sobre `n`.

Prova por indução

Caso base

```
length (replicate 0 x) = 0
```

Prova por indução

Caso base

```
length (replicate 0 x) = 0
```

```
length (replicate 0 x)
```

Prova por indução

Caso base

```
length (replicate 0 x) = 0
```

```
  length (replicate 0 x)
=    {replicate.1}
  length []
```

Prova por indução

Caso base

```
length (replicate 0 x) = 0
```

```
  length (replicate 0 x)
=    {replicate.1}
  length []
=    {length.1}
  0
```

Prova por indução (cont.)

Case indutivo

Hipótese: `length (replicate n x) = n`

Tese: `length (replicate (1+n) x) = 1+n`

Prova por indução (cont.)

Case indutivo

Hipótese: `length (replicate n x) = n`

Tese: `length (replicate (1+n) x) = 1+n`

```
length (replicate (1+n) x)
```

Prova por indução (cont.)

Case indutivo

Hipótese: `length (replicate n x) = n`

Tese: `length (replicate (1+n) x) = 1+n`

```
  length (replicate (1+n) x)
=    {replicate.2}
  length (x :  replicate n x)
```

Prova por indução (cont.)

Case indutivo

Hipótese: `length (replicate n x) = n`

Tese: `length (replicate (1+n) x) = 1+n`

```
  length (replicate (1+n) x)
=    {replicate.2}
  length (x :  replicate n x)
=    {length.2}
  1 + length (replicate n x)
```

Prova por indução (cont.)

Case indutivo

Hipótese: `length (replicate n x) = n`

Tese: `length (replicate (1+n) x) = 1+n`

```
  length (replicate (1+n) x)
=    {replicate.2}
  length (x :  replicate n x)
=    {length.2}
  1 + length (replicate n x)
=    {hipótese de indução}
  1 + n
```

Indução sobre listas

Também podemos provar propriedades usando indução sobre o comprimento das listas.

$$\frac{\begin{array}{l} P([\,]) \\ P(xs) \implies P(x : xs) \quad \text{para todo } x, xs \end{array}}{P(xs) \quad \text{para todo } xs}$$

Nota: propriedades de listas finitas!

Exemplo

Vamos mostrar que

```
xs ++ [] = xs
```

por indução sobre `xs`.

Exemplo

O caso base é trivial:

```
[] ++ [] = []          {++.1}
```

Exemplo

O caso base é trivial:

```
[] ++ [] = []        {++.1}
```

O caso indutivo é também simples.

Hipótese: `xs ++ [] = xs`

Tese: `(x:xs) ++ [] = (x:xs)`

Exemplo

O caso base é trivial:

```
[] ++ [] = []        {++.1}
```

O caso indutivo é também simples.

Hipótese: `xs ++ [] = xs`

Tese: `(x:xs) ++ [] = (x:xs)`

```
(x:xs) ++ []
```

Exemplo

O caso base é trivial:

```
[] ++ [] = []        {++.1}
```

O caso indutivo é também simples.

Hipótese: `xs ++ [] = xs`

Tese: `(x:xs) ++ [] = (x:xs)`

```
  (x:xs) ++ []
=    {++.2}
  x :  (xs ++ [])
```

Exemplo

O caso base é trivial:

```
[] ++ [] = []        {++.1}
```

O caso indutivo é também simples.

Hipótese: `xs ++ [] = xs`

Tese: `(x:xs) ++ [] = (x:xs)`

```
  (x:xs) ++ []
=    {++.2}
  x :  (xs ++ [])
=    {hipótese de indução}
  x:xs
```

Segundo exemplo

Mostrar

```
reverse (reverse xs) = xs
```

por indução sobre `xs`.

Segundo exemplo

Caso base:

```
reverse (reverse [])
```

Segundo exemplo

Caso base:

```
    reverse (reverse [])
=     {reverse.1 interior}
    reverse []
```

Segundo exemplo

Caso base:

```
    reverse (reverse [])
=     {reverse.1 interior}
    reverse []
=     {reverse.1}
    []
```

Segundo exemplo

Caso indutivo.

Hipótese: `reverse (reverse xs) = xs`

Tese: `reverse (reverse (x:xs)) = x:xs`

```
reverse (reverse (x:xs))
```

Segundo exemplo

Caso indutivo.

Hipótese: `reverse (reverse xs) = xs`

Tese: `reverse (reverse (x:xs)) = x:xs`

```
  reverse (reverse (x:xs))
=    {reverse.2 interior}
  reverse (reverse xs ++ [x])
```

Segundo exemplo

Caso indutivo.

Hipótese: `reverse (reverse xs) = xs`

Tese: `reverse (reverse (x:xs)) = x:xs`

```
  reverse (reverse (x:xs))
=    {reverse.2 interior}
  reverse (reverse xs ++ [x])
=
  ?
```

Necessitamos de um resultado auxiliar para continuar!

Dois lemas auxiliares

Distributividade de `reverse` sobre `++`

```
reverse (xs ++ ys) = reverse ys ++ reverse xs
```

Atenção à inversão da ordem dos argumentos!

Para provar o lema acima, necessitamos de mostrar:

Associatividade de `++`

```
(xs ++ ys) ++ zs = xs ++ (ys ++ zs)
```

Exercício: provar estes lemas usando indução.

De regresso à prova

```
  reverse (reverse (x:xs))
=    {reverse.2 interior}
  reverse (reverse xs ++ [x])
```

De regresso à prova

```
  reverse (reverse (x:xs))
=    {reverse.2 interior}
  reverse (reverse xs ++ [x])
=    {distributividade reverse/++}
  reverse [x] ++ reverse (reverse xs)
```

De regresso à prova

```
  reverse (reverse (x:xs))
=   {reverse.2 interior}
  reverse (reverse xs ++ [x])
=   {distributividade reverse/++}
  reverse [x] ++ reverse (reverse xs)
=   {reverse.2, reverse.1}
  [x] ++ reverse (reverse xs)
```

De regresso à prova

```
  reverse (reverse (x:xs))
=   {reverse.2 interior}
  reverse (reverse xs ++ [x])
=   {distributividade reverse/++}
  reverse [x] ++ reverse (reverse xs)
=   {reverse.2, reverse.1}
  [x] ++ reverse (reverse xs)
=   {hipótese de indução}
  [x] ++ xs
```

De regresso à prova

```
  reverse (reverse (x:xs))
=   {reverse.2 interior}
  reverse (reverse xs ++ [x])
=   {distributividade reverse/++}
  reverse [x] ++ reverse (reverse xs)
=   {reverse.2, reverse.1}
  [x] ++ reverse (reverse xs)
=   {hipótese de indução}
  [x] ++ xs
=   {++.2, ++.1}
  x:xs
```