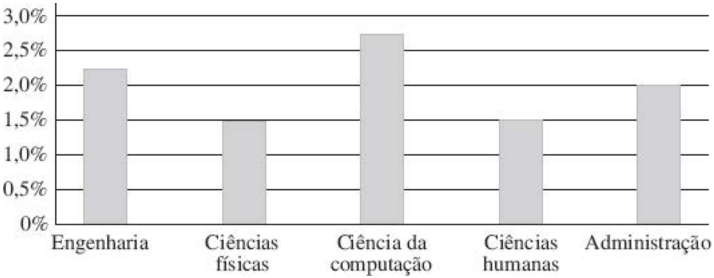
Funções

IPRJ01-10762
Matemática Discreta II

Gustavo Barbosa Libotte

Aula 5

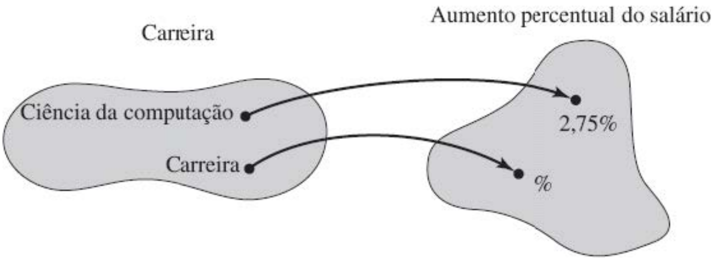
- Nesta aula discutiremos funções, que são, de fato, casos particulares de relações binárias de um conjunto S em um conjunto T.
- Essa visão de função no entanto é bastante sofisticada, e chegaremos a ela gradualmente.
- Função é uma palavra bastante comum mesmo em contextos não técnicos.
- Um jornal pode ter um artigo sobre como os salários iniciais para formandos deste ano cresceram em relação aos formandos do ano passado. Esse artigo poderia conter uma frase do tipo "O aumento do salário-base inicial depende da carreira" ou "O aumento do salário-base inicial é uma função da carreira".
- Essa relação funcional pode estar ilustrada por um gráfico, que mostra que cada carreira tem algum número para o aumento de salário associado e que tanto as ciências físicas quanto as ciências humanas têm o mesmo número, $1,5\%$.



- Também usamos, é claro, funções matemáticas em álgebra e cálculo. A equação $g(x) = x^3$ expressa uma relação funcional entre os valores de x e os valores correspondentes que resultam quando x é substituído por seu valor na expressão.

- Assim, um valor 2 para x tem o número $2^3 = 8$ associado a ele. (Esse número é expresso como $g(2) = 8$.)

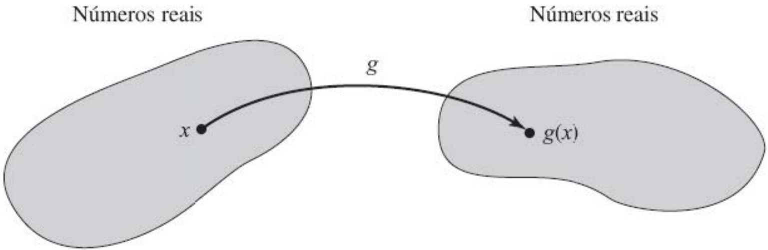
- Analogamente, $g(1) = 1^3 = 1, g(-1) = (-1)^3 = -1$, e assim por diante. Para cada valor de x, o valor correspondente para $g(x)$ é **único**.
- Se desenhássemos o gráfico dessa função em um sistema retangular de coordenadas, os pontos $(2,8), (1,1)$ e $(-1,-1)$ seriam pontos pertencentes ao gráfico.



- A função no exemplo sobre o aumento de salário pode ser descrita da seguinte maneira. Começamos com o diagrama na figura ao lado, que indica que a função sempre começa com uma dada carreira e que um aumento particular de salário está associado à carreira.

Funções

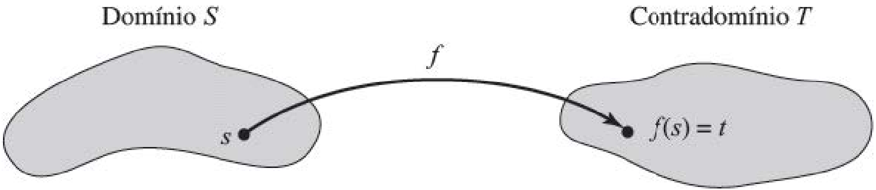
- A associação propriamente dita é descrita pelo conjunto de pares ordenados $\{($ engenharia; 2,25%), (ciências físicas; 1,5%), (ciências da computação; 2,75%), (ciências humanas; $1,5\%)$, (administração; 2,0%)$\}$.
- Para o exemplo algébrico $g(x) = x^3$, a figura a seguir mostra que a função sempre começa com um número real dado e tem um segundo número real associado a ele.
- A associação propriamente dita é descrita por $\{(x, g(x)) \mid g(x) = x^3\}$ ou simplesmente por $g(x) = x^3$.
- Esse conjunto inclui (2, 8), (1, 1) e (-1, -1), mas, como é um conjunto infinito, não podemos listar todos os seus elementos, temos que descrevê-los.



- Podemos concluir, dos exemplos acima, que existem três partes em cada função:
  1) um conjunto de valores iniciais;
  2) um conjunto do qual saem os valores associados;
  3) a associação propriamente dita.

- O conjunto de valores iniciais é chamado de **domínio da função**, e o conjunto do qual saem os valores associados é o **contradomínio da função**.

- Assim, tanto o domínio quanto o contradomínio representam conjuntos em que os valores são escolhidos.
- A figura a seguir mostra o desenho de uma função arbitrária f. Nessa figura, f é uma função de S em T, simbolizada por $f : S \to T$.
- S é o **domínio** e T, o **contradomínio**. A associação propriamente dita é um conjunto de pares ordenados da forma (s, t), em que $s \in S, t \in T$ e t é o valor em T que a função associa ao valor s em $S; t = f(s)$.
- Logo, a associação é um subconjunto de $S \times T$ (uma **relação binária** de S em T).
- Mas a propriedade importante dessa relação é que todo elemento de S tem que ter associado a si um e um único valor em T, de modo que todo $s \in S$ aparece exatamente uma vez como a primeira componente de um par (s, t).
- Essa propriedade não evita que um dado valor em T apareça mais de uma vez.



Terminologia para funções

Sejam S e T conjuntos. Uma função (aplicação) f de S em T,f. $S \to T$,é um subconjunto de $S \times T$ tal que cada elemento de S aparece exatamente uma vez como a primeira componente de um par ordenado. S é o **domínio** e T é o **contradomínio** da função. Se (s, t) pertencer à função, então denotaremos t por $f(s)$; t é a **imagem** de s sob f, s é uma **imagem inversa** de t sob f e f leva s em t. Para $A \subseteq S$,$f(A)$ denota $\{f(a) \mid a \in A\}$.

- Uma função de S em T é um subconjunto de $S \times T$ contendo algumas restrições sobre os pares ordenados que contém.
- É por isso que falamos de uma função como um tipo particular de relação binária. Pela definição de função, uma relação do tipo um para muitos (ou muitos para muitos) não pode ser uma função.
- Além disso, cada elemento de S tem que aparecer como uma primeira componente.
- Falamos muito sobre valores nos conjuntos S e T, mas, como mostra nosso exemplo sobre aumento de salário, esses valores não são necessariamente números, nem a associação propriamente dita é descrita necessariamente por uma equação.
- A definição de função inclui funções de mais de uma variável. Podemos ter uma função $f : S_1 \times S_2 \times \ldots \times S_n \to T$ que associa a cada n-upla de elementos $(s_1, s_2, . \quad \ldots, s_n), s_i \in S$, um único elemento de T.

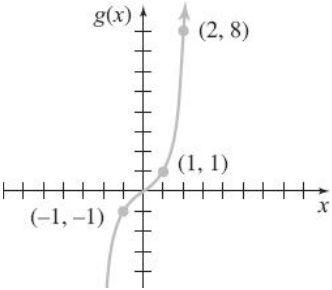
- A definição de uma função $f : S \to T$ tem três partes—o domínio S, o contradomínio T e a associação propriamente dita. Tudo isso é necessário?
- **Por que não podemos, simplesmente, escrever uma equação, como $g(x) = x^3$, para definir uma função?**
- A resposta mais rápida é que nem todas as associações funcionais podem ser descritas por uma equação.

Exemplo

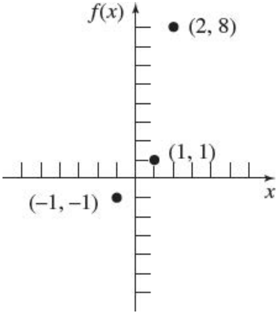
Seja S o conjunto de todas as cadeias de caracteres de comprimento finito. Então, a associação que leva cada cadeia em seu número de caracteres é uma função com domínio S e contradomínio N (permitimos a "cadeia vazia", que tem zero caracteres).

- Mas tem mais—vamos limitar nossa atenção a situações em que é possível usar uma equação para descrever a associação, como $g : R \to R$, em que $g(x) = x^3$. Mesmo em álgebra e cálculo é comum dizer "considere uma função $g(x) = x^3$", implicando que a equação é a função.

- Tecnicamente, a equação descreve apenas **um modo** de se calcular os valores associados.
- A função $h : R \to R$ dada por $h(x) = x^3 - 3x + 3(x+5) - 15$ é a mesma função que g, já que contém os mesmos pares ordenados.
- No entanto, a equação é diferente, no sentido em que processa qualquer valor dado de x de maneira diferente.
- Por outro lado, a função $f : Z \to R$ dada por $f(x) = x^3$ é uma função diferente de g.
- O domínio foi modificado, o que muda o conjunto de pares ordenados.
- O gráfico de f é formado por pontos discretos. Mesmo em situações em que uma quantidade varia continuamente em relação a uma outra, em um computador digital aproximamos pegando dados discretos em intervalos pequenos, como o gráfico da função $g(x)$ é aproximado pelo gráfico de $f(x)$.



(a) $g(x)$ contínua



(a) $f(x)$ discreta

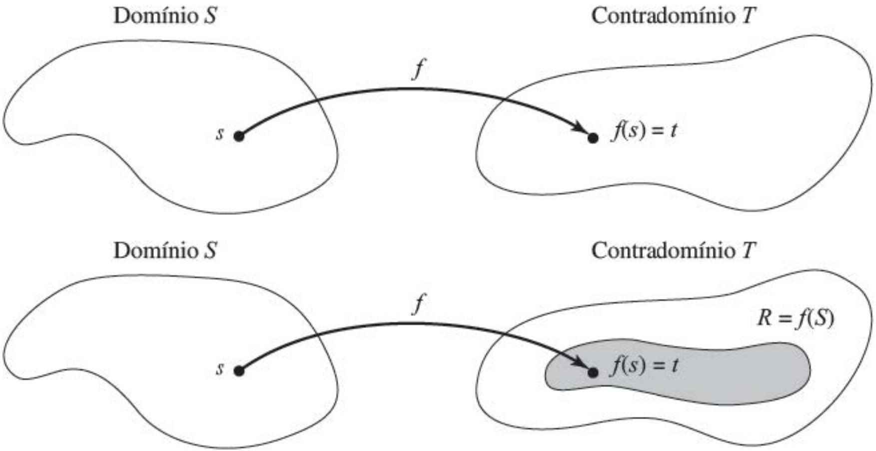
- Finalmente, vamos olhar para a função $k : R \to C$ dada por $k(x) = x^3$. A equação e o domínio são os mesmos que para $g(x)$; o contradomínio foi aumentado, mas isso não afeta os pares ordenados.
- Vamos considerar essa função como igual a $g(x)$?
- Ela não é igual, mas, para ver por quê, temos que esperar até a discussão de funções sobrejetoras.
- Veremos, então, que g é sobrejetora, enquanto k não é, de modo que não queremos considerá-las como a mesma função.

Funções sobrejetoras

- Seja $f : S \to T$ uma função arbitrária com domínio S e contradomínio T.
- Parte da definição de uma função é que todo elemento de S tem uma imagem sob f e que todas as imagens são elementos de T; o conjunto R de todas as imagens é chamado de imagem da função f.
- Assim, $R = \{f\ (\mathrm{s}) \mid s \in S\}$, ou $R = f(S)$. É claro que $R \subseteq T$;
- Se acontecer que $R = T$, ou seja, que a imagem coincide com o contradomínio, então a função é dita sobrejetora.

Função sobrejetora

Uma função $f : S \to T$ é dita sobrejetora (ou sobrejetiva) se sua imagem for igual a seu contradomínio.



A imagem R está sombreada.

- Para toda função com imagem R e contradomínio T, temos $R \subseteq T$.
- Para mostrar que determinada função é sobrejetora, precisamos mostrar que $T \subseteq R$; teremos, então, $R = T$.
- Precisamos mostrar, portanto, que um elemento arbitrário do contradomínio pertence à imagem, ou seja, é a imagem de algum elemento no domínio.
- Por outro lado, se pudermos produzir um elemento no contradomínio que não é imagem de nenhum elemento do domínio, teremos provado que a função não é sobrejetora.
- **Em resumo**: para mostrar que uma função é sobrejetora, pegue um elemento arbitrário no contradomínio e mostre que ele tem uma imagem inversa no domínio.

Exemplo

- A função $g : R \to R$ definida por $g(x) = x^3$ é sobrejetora.
- Para provar isso, seja r um número real arbitrário e seja $x = \sqrt[3]{r}$.
- Então x é um número real, de modo que x pertence ao domínio de g e $g(x) = (\sqrt[3]{r})^3 = r$.
- Logo, qualquer elemento do contradomínio é a imagem, sob g, de um elemento do domínio.
- A função $k : R \to \mathbb{C}$ dada por $k(x) = x^3$ não é sobrejetora.
- Existem muitos números complexos (i, por exemplo) que não podem ser obtidos elevando-se ao cubo um número real.
- Portanto, g e k não são funções iguais.

Funções injetoras

- A definição de função garante que existe uma única imagem para cada elemento do domínio.
- No entanto, determinado elemento da imagem pode ter mais de uma imagem inversa.
- No nosso primeiro exemplo de função (aumento do salário inicial), tanto as ciências físicas quanto as humanas tinham imagem inversa iguais a 3%.
- Essa função não era injetora.

Função injetora

Uma função $f : S \to T$ é dita injetora (ou injetiva ou um para um) se nenhum elemento de T é a imagem, sob f de dois elementos distintos em S.

- **Em resumo**: para mostrar que uma função f é injetora, suponha que $f(s1) = f(s2)$ e mostre que $s1 = s2$.

Exemplo

- A função $g : R \to R$ definida por $g(x) = x^3$ é injetora, pois, se x e y são números reais com $g(x) = g(y)$, então $x^3 = y^3$ e $x = y$.
- A função $f : R \to R$ dada por $f(x) = x^2$ não é injetora, pois $f(2) = f(-2) = 4$, por exemplo.
- No entanto, a função $h : N \to N$ dada por $h(x) = x^2$ é injetora porque, se x e y são inteiros não negativos com $h(x) = h(y)$, | então $x^2 = y^2$, e, como x e y são ambos não negativos, $x = y$.

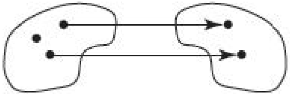
Bijeções

Função bijetora

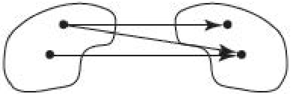
Uma funçãof: $S \to T$ é bijetora (ou bijetiva ou uma bijeção) se é, ao mesmo tempo, injetora e sobrejetora.

Exemplo

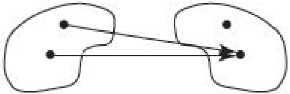
A função g: $R \to R$ dada por $g(x) = x^3$ é uma bijeção. A função $f : R \to R$ dada por $f\ (x) = x^2$ não é uma bijeção (não é injetora), assim como a função $k : R \to \mathbb{C}$ dada por $k(x) = x^3$ (não é sobrejetora).



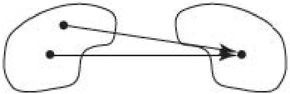
Não é função



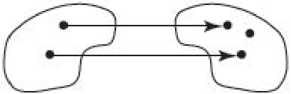
Não é função



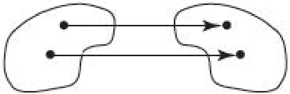
Função, não é injetora nem sobrejetora



Função, não é injetora,
mas é sobrejetora



Função, injetora,
não é sobrejetora

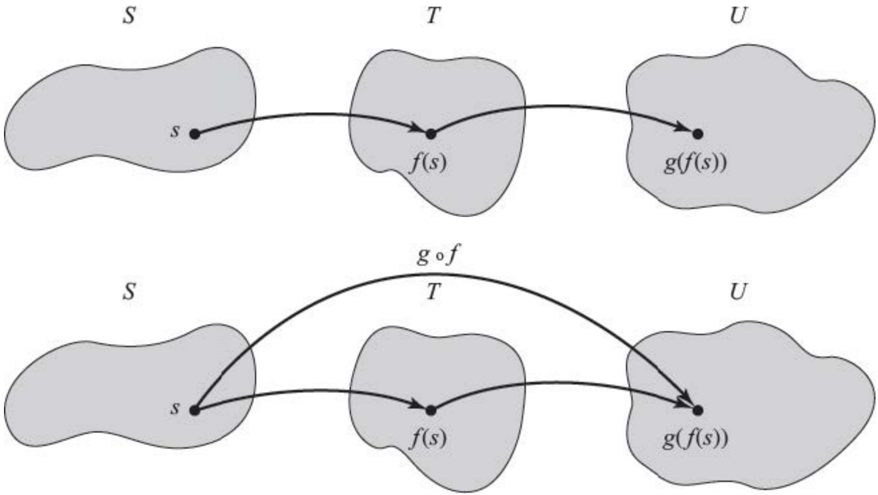


Função, injetora
e sobrejetora

- Suponha que f e g são funções, com $f : S \to T$ e $g : T \to U$.
- Então, para qualquer $s \in S, f(s)$ é um elemento de T, que é o domínio de g.
- Logo, a função g pode ser calculada em $f(s)$. O resultado é $g(f(s))$, um elemento de U.
- Escolher um elemento arbitrário s de S, aplicar a função f e depois aplicar a função g a $f(s)$ é o mesmo que associar um único elemento de U a s.
- Resumindo, criamos uma função $S \to U$, chamada a composição das funções f e g e denotada por $g \circ f$.

Função composta

Sejam $f : S \to T$ e g: $T \to U$. A função composta $g \circ f$ é a função de S em U definida por $(g \circ f)(s) = g(f(s))$.


S
T
U
s
f(s)
g(f(s))
g ∘ f
S
T
U
s
f(s)
g(f(s))

$$\begin{tikzcd} S \arrow[r, "f"] \arrow[dr, "g \circ f"'] & T \arrow[d, "g"] \\ & U \end{tikzcd}$$

- O diagrama na figura ao lado ilustra, também, a definição de função composta.
- As letras nas quinas indicam os domínios e contradomínios das três funções.
- O diagrama diz que, começando com um elemento de S, se seguirmos o caminho indicado por $g \circ f$ ou o caminho indicado por f seguido do caminho indicado por g, obtemos o mesmo elemento de U.
- Diagramas ilustrando que caminhos alternativos produzem o mesmo efeito são chamados de diagramas comutativos.

- Nem sempre é possível fazer a composição de duas funções arbitrárias; os domínios e as imagens têm que ser "compatíveis".

- Por exemplo, se $f : S \to T$ e $g : W \to Z$, em que T e W são disjuntos, então $(g \circ f)(s) = g(f(s))$ não está definida, pois $f(s)$ não pertence ao domínio de g. **Lembrete**: note que a função $g \circ f$ é aplicada da direita para a esquerda: primeiro aplicamos f e depois g.

Exercícios

Considere as funções f: $A \to B$ e g: $B \to C$, onde $A = \{1, 2, 3, 4\}, B = \{a, b, c, d\}$ e $C = \{x, y, z\}$. Mostre se cada uma das seguintes funções é injetora, sobrejetora ou bijetora:

a) $f(1) = a, f(2) = b, f(3) = c, f(4) = d$;

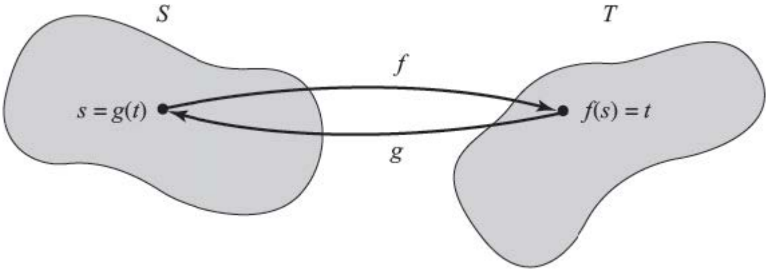
b) $g(a) = x, g(b) = y, g(c) = y, g(d) = z$;

c) $g(f(1)) = x, g(f(2)) = y, g(f(3)) = z, g(f(4)) = x$

- Funções bijetoras têm outra propriedade importante. Seja $f : S \to T$ uma bijeção.
- Como f é sobrejetora, todo $t \in T$ tem uma imagem inversa em S. Como f é injetora, essa imagem inversa é única.
- Podemos então associar a cada elemento $t \in T$ um único elemento em S, a saber, $s \in S$, tal que $f(s) = t$.
- Essa associação descreve uma função $g : T \to S$. Os domínios e contradomínios de f e g são tais que podemos formar tanto $g \circ f : S \to S$ quanto $f \circ g : T \to T$.
- Se $s \in S$, então $(g \circ f)(s) = g(f(s)) = g(t) = s$. Logo, $g \circ f$ leva cada elemento de S em si mesmo.
- A função que leva cada elemento de um conjunto S em si mesmo, ou seja, que deixa cada elemento de S fixo, é chamada de função identidade em S e denotada por i_S.
- Portanto, $g \circ f = i_S$.

Função inversa

Seja f uma função, $f : S \to T$. Se existir uma função $g : T \to S$ tal que $g \circ f = i_S$ e $f \circ g = i_T$, então g é chamada a função inversa de f denotada por f^{-1}.



Exercícios

1. Considere a função $f : \mathbb{R} \to \mathbb{R}$ definida por $f(x) = 2x - 3$. Encontre a função inversa de f e mostre que ela é realmente a inversa de f.
2. Considere a função $f : [0, \infty) \to [0, \infty)$ definida por $f(x) = x^2 + x + 1$. Verifique se a função f é bijetora e determine sua função inversa.

- A ordem de grandeza é uma maneira de comparar a “taxa de crescimento” de funções diferentes.
- Sabemos, por exemplo, que se calcularmos $f(x) = x$ e $g(x) = x^2$ para valores cada vez maiores de x os valores de g serão maiores do que os valores de f, e a diferença é cada vez maior.
- Essa diferença na taxa de crescimento não vai deixar de existir se simplesmente multiplicarmos os valores de f por uma constante muito grande; não importa quão grande for a constante que escolhermos, os valores de g certamente começarão a ficar cada vez maiores do que os de f.

- Nossa experiência indica que as funções f e g se comportam de maneiras fundamentalmente diferentes em relação às suas taxas de crescimento.
- Para caracterizar essa diferença formalmente, definiremos uma relação binária nas funções.
- Seja S o conjunto de todas as funções com domínio e contradomínio iguais aos números reais não negativos.
- Podemos definir uma relação binária em S por $f \; \rho \; g \leftrightarrow$ existem constantes positivas n_0, c_1 e c_2 tais que para todo $x \geq n_0$, $c_1 g(x) \leq f(x) \leq c_2 g(x)$.

Exemplo

Sejam f e g funções em S onde $f(x) = 3x^2$ e $g(x) = 200x^2 + 140x + 7$. Seja $n_0 = 2, c_1 = \frac{1}{100}$ e $c_2 = 1$. Então para $x \geq 2$,

$$\frac{1}{100}\left(200x^2 + 140x + 7\right) \leq 3x^2 \leq (1)\left(200x^2 + 140x + 7\right)$$

Exemplo (continuação)

ou ainda

$$2x^2 + 1,4x + 0,07 \leq 3x^2 \leq 200x^2 + 140x + 7$$

Portanto $f \ \rho \ g$.

- A relação ρ é uma **relação de equivalência** em S. Por exemplo, para provar que $f \ \rho \ f$, escolhemos $n_0 = c_1 = c_2 = 1$ e obtemos
$$(1)f(x) \leq f(x) \leq (1)f(x)$$
- Dado que r é uma relação de equivalência, ela divide S em **classes de equivalência**.
- Se f está na mesma classe que g, então dizemos que f tem a mesma ordem de grandeza do que g, denotamos por $f = \Theta(g)$ e lemos "f é da ordem de g".

- Devido à simetria, isso significa também que g tem a mesma ordem de grandeza que f, ou $g = \Theta(f)$.[1]

Ordem de grandeza

Sejam f e g funções dos reais não negativos nos reais não negativos. Então f tem a mesma ordem de grandeza do que g, denotado por $f = \Theta(g)$, se existem constantes positivas n_0, c_x e c_2 tais que, se $x \geq n_0$, então $c_1 g(x) \geq f(x) \geq c_2 g(x)$.

- Em geral, tentaremos encontrar o representante mais simples de determinada classe de equivalência. Assim, para as funções $f(x) = 3x^2$ e $g(x) = 200x^2 + 140x + 7$ no exemplo anterior, diríamos que $f = \Theta\left(x^2\right)$ e que $g = \Theta\left(x^2\right)$.

[1] A notação $f = \Theta(g)$ é um abuso de notação, já que $\Theta(g)$ não é alguma função idêntica a f. É, simplesmente, uma abreviação do fato de que $f \in [g]$ sob a relação de equivalência r definida anteriormente.

- Um polinômio **sempre tem a mesma ordem de grandeza** do seu monômio de maior grau; os monômios de menor grau e todos os coeficientes **podem ser ignorados**.
- Isso não é nenhuma surpresa, já que o monômio de maior grau dominará o resultado.
- Para compreender de maneira mais intuitiva o significado dessas classes de equivalência, vamos desenhar alguns gráficos.
- Seja $h(x) \in S$, em que $h(x) = x^2$. A figura (a) ao lado mostra o gráfico de $h(x)$.

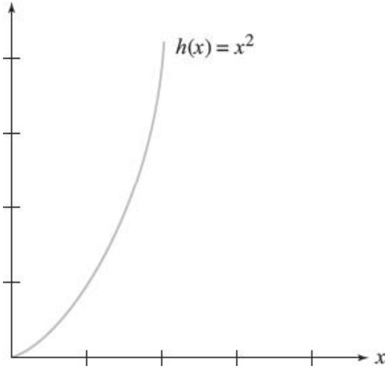


figura (a)

- Suponha, agora, que multiplicamos os valores de h por duas constantes, $c_1 = 1/2$ e $c_2 = 2$.
- As funções $c_1 h(x)$ e $c_2 h(x)$ aparecem na figura (b) ao lado, como linhas tracejadas.
- Essas linhas tracejadas formam uma espécie de envelope em torno dos valores de $h(x)$, tendo, aproximadamente, a mesma forma que $h(x)$.
- Uma mudança nos valores das constantes muda a largura do envelope, mas não a forma básica.
- Se $h_1(x)$ é uma função tal que $h_1(x) = \Theta(h)$, então existem uma constante positiva n_0 e algum envelope em torno de $h(x)$ tal que, para todos os valores do domínio à direita de n_0, os valores de h_1 caem dentro desse envelope.

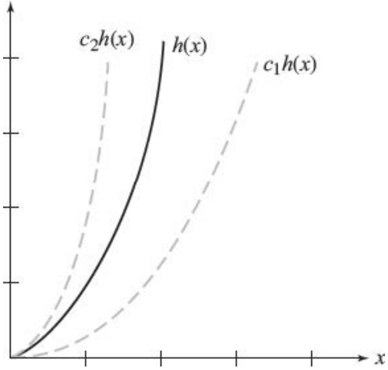


figura (b)

Instituto Politécnico

- Este fato está ilustrado na figura (c) ao lado.
- Portanto, os valores de h_1 nunca podem ficar longe demais dos valores de h.
- As funções h_1 e h são, aproximadamente, do mesmo "tamanho"—elas têm a **mesma ordem de grandeza**.
- Se imaginarmos funções representando diversos meios de transporte, então funções de mesma ordem de grandeza (pertencentes à mesma classe de equivalência) representam o mesmo meio de transporte.
- Uma classe representa andar a pé, outra, andar de carro, uma terceira, de avião.
- As velocidades dentro de uma mesma modalidade são aproximadamente iguais.

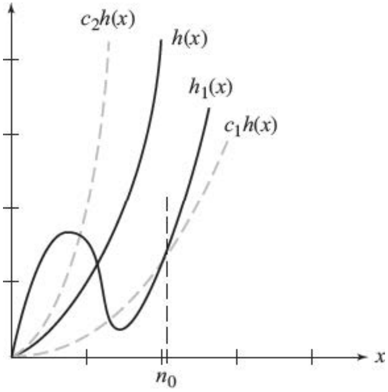


figura (c)

- Ignorar os coeficientes e os monômios de menor grau é o mesmo que ignorar a diferença entre andar e correr, ou entre um jipe e um Jaguar, ou entre um Cessna e um Boeing 787.
- Andar (a qualquer velocidade) é **muito diferente** de dirigir, que é **muito diferente** de voar.
- Podemos imaginar uma hierarquia de ordens de grandeza. Por exemplo, a classe $\Theta(x)$ tem ordem de grandeza menor do que $\Theta\left(x^2\right)$, pois as funções que são $\Theta(x)$ acabam ficando abaixo das que são $\Theta\left(x^2\right)$.
- Além disso, a classe $\Theta(\log x)$ tem ordem de grandeza menor do que $\Theta(x)$.
- Na nossa analogia sobre meios de transporte, andar é mais lento do que dirigir, que é mais lento do que voar.

- A ordem de grandeza é importante na análise de algoritmos. Ao analisar um algoritmo, identificamos as tarefas importantes executadas por ele.
- **Em geral, o número de vezes que tais tarefas serão executadas depende do tamanho dos dados de entrada.**
- Por exemplo, procurar em uma lista de n elementos ou ordenar uma lista de n elementos vai ficar cada vez mais trabalhoso à medida que n cresce.
- Tipicamente, podemos expressar o tamanho dos dados de entrada como um inteiro não negativo, de modo que as funções que expressam a quantidade de trabalho vão ter domínio $\mathbb{N}$.
- Uma busca sequencial entre n elementos necessita de n comparações no pior caso, enquanto uma busca binária precisa de $1 + \log n$ comparações no pior caso (supondo que n é uma potência de 2).
- Em vez de calcular as funções exatas para a quantidade de trabalho executado, é mais fácil, e muitas vezes tão útil quanto, usar a informação sobre ordem de grandeza.

ALGORITMO	OPERAÇÃO	ORDEM DE GRANDEZA DO PIOR CASO
BuscaSequencial em uma lista de tamanho n	comparações	$\Theta(n)$
BuscaBinária em uma lista ordenada de tamanho n	comparações	$\Theta(\log n)$
Busca de um padrão de comprimento m em um texto de comprimento n	comparações	$\Theta(mn)$
Cálculo de um polinômio de grau n	multiplicações e somas	$\Theta(n)$
OrdenaçãoPorBolhas de uma lista de tamanho n	comparações	$\Theta\left(n^2\right)$
OrdenaçãoPorSeleção de uma lista de tamanho n	comparações	$\Theta\left(n^2\right)$

- Busca sequencial é $\Theta(n)$ e busca binária é $\Theta(\log n)$ no pior caso.

- Assim, a busca binária é uma melhora em ordem de grandeza sobre a busca sequencial.
- A ordem de grandeza de um algoritmo também é conhecida como sua complexidade computacional, pois reflete a quantidade de trabalho inerente ao algoritmo.
- Para entender o efeito da ordem de grandeza na análise de algoritmos, suponha que temos dois algoritmos, A e A', que fazem a mesma coisa, mas diferem em ordem de grandeza; digamos que A é $\Theta(n)$ e que A' é $\Theta\left(n^2\right)$.
- Mesmo que cada passo nos cálculos leve apenas 0,0001 segundo, essa diferença vai afetar o tempo total de computação quando n vai ficando maior.
- As duas primeiras linhas da tabela a seguir fornecem o tempo total de computação de A e de A' para diversos valores de tamanho dos dados.
- Suponha, agora, que existe um terceiro algoritmo A' cuja ordem de grandeza não é polinomial, mas uma função exponencial, digamos 2^n.

		TAMANHO DOS DADOS DE ENTRADA (n)		
Algoritmo	**Ordem**	**10**	**50**	**100**
A	n	0,001 segundo	0,005 segundo	0,01 segundo
A'	n^2	0,01 segundo	0,25 segundo	1 segundo
A'	2^n	0,1024 segundo	3570 anos	4×10^{16} séculos

- Observe que o caso exponencial cresce a uma taxa fantástica!
- Mesmo supondo que cada cálculo leva muito menos do que 0,0001 segundo, as taxas de crescimento relativas entre as funções polinomial e exponencial ainda seguem esse mesmo padrão.
- Devido a essa imensa taxa de crescimento, algoritmos que não são de ordem polinomial são, em geral, inúteis para valores grandes de n.

- De fato, problemas para os quais não existe algoritmo em tempo polinomial são ditos **intratáveis**.
- Algumas vezes algoritmos que não são polinomiais no pior caso ainda podem ser eficientes—e úteis—em casos de dados de entrada "médios".
- Não obstante, ao tentar melhorar a eficiência, deveríamos perguntar se existe um algoritmo diferente com ordem de grandeza menor, antes de nos preocuparmos com os detalhes de ajuste fino para um algoritmo dado.
- Se $f(n)$ representar o trabalho executado por um algoritmo com dados de entrada de tamanho n, poderá ser difícil encontrar uma função simples g tal que $f = \Theta(g)$.
- Lembre-se de que, se encontrarmos tal g, f e g são funções que acabarão ficando (para n suficientemente grande) com aproximadamente a mesma forma.

- Mas podemos ainda ser capazes de encontrar uma função g que sirva como uma cota superior para f. Em outras palavras, embora f possa não ter a mesma forma que g, f nunca vai crescer significativamente mais rápido do que g.
- Formalmente, isso é expresso por $f = O(g)$ ("f é O grande de g").

"O grande (*big-O*)"

Sejam f e g funções dos reais não negativos nos reais não negativos. Então f será "O grande" de g, denotado por $f = O(g)$, se existirem constantes positivas n_0 e c tais que para $x \geq n_0, f(x) \leq cg(x)$.

- A notação Ogrande $f = O(g)$ diz que f cresce à mesma taxa ou a uma taxa menor do que g.
- Se $f = O(g)$, então g é uma cota superior para f e nos dá uma ilustração do pior caso para o crescimento de f.

Instituto Politécnico

A função módulo

Para qualquer inteiro x e qualquer inteiro positivo n, a função módulo n, denotada por $f(x) = x \bmod n$, associa a cada x o resto de sua divisão por n. Podemos escrever

$$x = q \cdot n + r\ ,$$

$0 \leq r < n$, em que q é o quociente e r é o resto, de modo que o valor de $x \bmod n$ é r.

- Veja os exemplos abaixo:
  - **a)** $25 = 12 \cdot 2 + 1$, logo $25 \bmod 2 = 1$
  - **b)** $21 = 3 \cdot 7 + 0$, logo $21 \bmod 7 = 0$
  - **c)** $15 = 3 \cdot 4 + 3$, logo $15 \bmod 4 = 3$
- Em outras palavras, $x \bmod n$ será o resto não negativo (também chamado de resíduo de x módulo n) quando x for dividido pelo inteiro positivo n.
- Essa função aparentemente “inofensiva” tem um número surpreendente de aplicações.

- Uma **função de dispersão** é uma função $h : S \to T$ tal que o domínio S é um conjunto de cadeias de texto ou valores inteiros e o contradomínio T é o conjunto de inteiros $\{0, 1, \ldots, t-1\}$, em que t é algum inteiro positivo relativamente pequeno.
- Se o domínio S consistir em cadeias de textos, podemos imaginá-las codificadas de alguma forma como valores inteiros, talvez por um algoritmo tão simples como o que converte cada letra individual em uma cadeia de texto em sua posição no alfabeto ($a = 1, b = 2$ e assim por diante) e depois soma a lista de inteiros resultante para obter um valor inteiro.
- Podemos, então, começar supondo que S consiste em valores inteiros.
- A função h, portanto, leva um conjunto possivelmente grande de inteiros S em uma janela relativamente pequena de valores inteiros T.
- Logo, h **não deve ser uma função injetora**, já que podem existir muitos valores $x_1, x_2, x_3, \ldots$ em S tais que $h(x_1) = h(x_2) = h(x_3)$.

- Nesse caso, dizemos que o valor da função foi "dispersado" para x_1, x_2, x_3.
- Seja o que for que a função $h(x)$ faça com x, o último passo é quase sempre aplicar a função mod t para que o resultado final caia no contradomínio $\{0, 1, \ldots, t-1\}$.

Exemplo

Sé um conjunto de inteiros positivos, T é o conjunto de inteiros $\{0, 1, \ldots, t-1\}$ e a função de dispersão h é dada por

$$h(x) = x \bmod t$$

Se $t = 10$, por exemplo, então os valores são calculados módulo 10:

$$h(7) = 7 \bmod 10 = 7$$
$$h(23) = 23 \bmod 10 = 3$$
$$h(59) = 59 \bmod 10 = 9$$

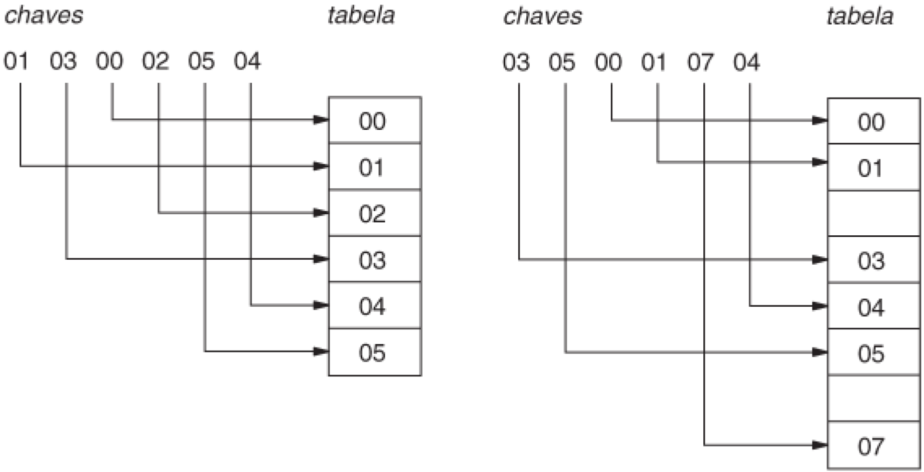
$$h(158) = 158 \bmod 10 = 8$$
$$h(48) = 48 \bmod 10 = 8$$

Aqui o valor 8 foi dispersado para 158 e 48.

Hashing

- Uma função de dispersão é usada com frequência em um **algoritmo de busca**.
- Uma solução bem diferente das vistas até agora para resolver os problemas de busca em uma tabela é a **tabela de dispersão**—também conhecida como tabela de espalhamento ou *hashing*.
- Ao invés de organizar a tabela segundo o valor relativo de cada chave em relação às demais, a tabela de dispersão leva em conta somente o seu valor absoluto, interpretado como um valor numérico.
- Através da aplicação de uma função conveniente, **a chave é transformada em um endereço de uma tabela**.
- A intenção é atingir diretamente, se possível, o local onde a chave se encontra.
- Na realidade, o método aproveita a possibilidade de acesso randômico à memória para alcançar uma complexidade média por operação de $O(1)$, sendo o pior caso, entretanto, $O(n)$.

- Suponha que existam n chaves a serem armazenadas em uma tabela T, sequencial e de dimensão m.
- As posições da tabela se situam no intervalo $[0, m-1]$. Isto é, a tabela é particionada em m compartimentos, cada um correspondendo a um endereço e podendo armazenar r nós distintos.
- O objetivo é armazenar cada chave no bloco referente a seu endereço. A busca, assim, requer somente **um acesso a um bloco**.
- O desenvolvimento das técnicas de tabelas de dispersão foi motivado por um caso bastante simples, porém importante.
- Suponha que o número de chaves n seja igual ao número de compartimentos m. Além disso, que os valores das chaves sejam, respectivamente, $0, 1, \ldots, m-1$.
- Pode-se então fazer o acesso direto ao valor de cada chave como seu índice na tabela; isto é, cada chave x é armazenada no compartimento x.

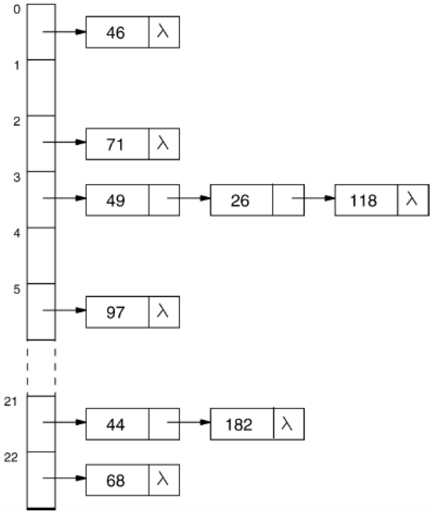

chaves
tabela
01 03 00 02 05 04
00
01
02
03
04
05
chaves
tabela
03 05 00 01 07 04
00
01
03
04
05
07

- Para o tratamento do caso geral, sem perda de generalidade, considera-se todas as chaves como numéricas, pois todo dado não numérico corresponde a uma representação numérica no computador.
- Com isso, por que não utilizar a técnica de acesso direto mesmo com espaços vazios? A resposta é simples.
- A quantidade de espaços vazios pode ser proibitiva. Como exemplo extremo, seja um conjunto de duas chaves apenas, de valores 0 e 999.999, respectivamente.
- A aplicação da técnica de acesso direto conduziria a uma tabela com 1.000.000 compartimentos, dos quais apenas dois ocupados.
- Para resolver essa questão, é possível transformar cada chave x em um valor no intervalo $[0, m-1]$ através da aplicação de uma **função de dispersão** h.

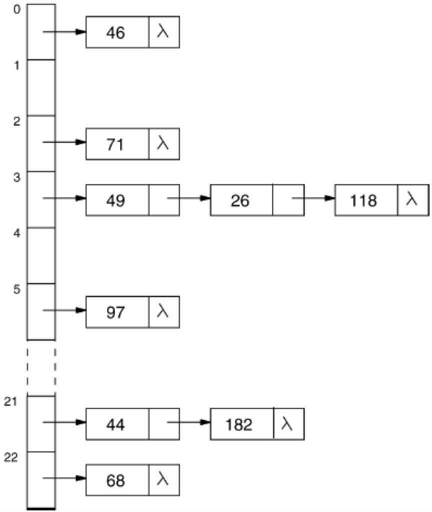
- Dada a chave x, determina-se o valor $h(x)$, denominado endereço-base de x. Se o compartimento $h(x)$ estiver desocupado, poderá ser utilizado para armazenar a chave x.
- Infelizmente, a função de dispersão **pode não garantir injetividade**, pois é possível a existência de outra chave $y \neq x$, tal que $h(y) = h(x)$.
- Esse fenômeno é denominado **colisão**, e as chaves x e y são **sinônimas** em relação a h.
- Na ocorrência desse fato, utiliza-se um procedimento especial para o armazenamento de x, denominado **tratamento de colisões** (ou tratamento de sinônimos).
- A seguir vamos explorar duas formas para o tratamento de colisões. Existem outras formas para realizar esta operação.

Instituto Politécnico

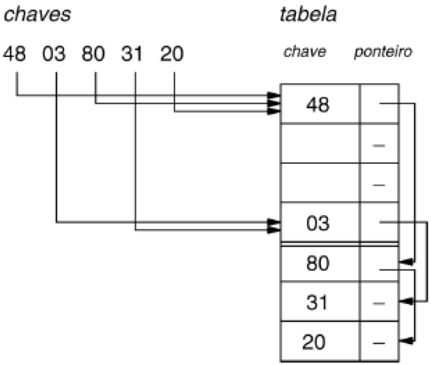
- O número médio de comparações necessárias para se buscar um elemento usando dispersão não depende do número n de elementos na tabela de dispersão, mas depende da razão entre n e o tamanho total m da tabela.
- Define-se como **fator de carga** de uma tabela de dispersão o valor $\alpha = n/m$, onde n é o número de chaves armazenadas.
- Na busca sequencial ou na busca binária, quanto maior for o número de elementos no conjunto em que é efetuada a busca, maior será o trabalho (número de comparações) necessário.
- Mas com o uso de dispersão podem ser buscados muitos ou poucos elementos com a mesma eficiência, desde que o fator de carga seja baixo.
- Um método para diminuir colisões é então **reduzir o fator de carga**; à medida que este cresce, a possibilidade de ocorrerem colisões também cresce.



- Uma nova chave sempre pode encontrar o seu endereço-base já ocupado.
- Por esse motivo, o emprego de tabelas de dispersão implica, necessariamente, a previsão de algum método de tratamento de colisões.
- O **encadeamento exterior** é uma solução muito usada para o problema, e consiste em manter m listas encadeadas, uma para cada possível endereço-base.



- Um campo para o encadeamento deve ser acrescentado a cada nó.
- Os nós correspondentes aos endereços-base serão apenas nós-cabeça para essas listas.
- A figura ao lado apresenta ilustrativamente esta solução.
- A complexidade do pior caso é simples; como o comprimento de uma lista encadeada pode ser $O(n)$, esta será a complexidade do pior caso, visto que pode ser necessário percorrer a lista até o final.



- O encadeamento interior prevê a divisão da tabela T em duas zonas: uma de endereços-base, de tamanho p, e outra reservada aos sinônimos, de tamanho s.
- Naturalmente, $p + s = m$. Os valores p e s são fixos.
- Assim sendo, a função de dispersão deve obter endereços-base na faixa $[0, p-1]$ apenas. Nesse caso, $\alpha = n/m \leq 1$.
- A estrutura da tabela é a mesma que no caso do encadeamento exterior.
- Dois campos têm presença obrigatória em cada nó. O primeiro é reservado ao armazenamento da chave, enquanto o segundo contém um ponteiro que indica o próximo elemento da lista de sinônimos correspondentes ao endereço-base em questão.