Teoria da Computação
Pode ser dividida em duas grandes áreas:
Computabilidade - Estudo do que é possível fazer com o computador, buscando determinar os limites da computação algorítmica (quais problemas que podem ou nao ser resolvidos por uma máquina de Turing)
Complexibilidade - Estuda quanto um problema é fácil ou difícil de ser resolvido por um computador, separando os problemas:

- Decidíveis
    - Tratáveis - Admite uma solução computacional em um tempo de execução razoável.
    - Intratáveis - Possui uma solução, entretanto, leva muito tempo para ser executada.
- Indecidíveis

Teoria da Parada
Na teoria da computabilidade o experimento mental do problema da parada é um problema de decisão que pode ser declarado informalmente da seguinte forma: "Dadas uma descrição de um programa e uma entrada finita, decida se o programa termina de rodar ou rodar indefinidamente.", que se torna um paradoxo.

Revisão de alguns termos da Matemática Discreta
Lida com conjuntos discretos (ex: números naturais e inteiros) e Funções:

Funções

- Mapeia os elementos do domínio em elementos do contradomínio, e os elementos que foram mapeados formam a imagem da função
- Podem ser injetoras, sobrejetoras e bijetoras
    - Injetoras - Mapeia elementos distintos do domínio e do contradomínio (assim não terá duas ou mais setas apontando para o mesmo elemento);
    - Sobrejetoras - Mapeia todos os elementos do contradomínio é o conjunto imagem é igual ao contradomínio (assim nunca haverá um elemento do contradomínio sem uma seta incidente);
    - Bijetoras - É uma função injetora e sobrejetora ao mesmo tempo, ou seja, há correspondência de pares no domínio e contradomínio (é igual a imagem)
- Função total: $X \rightarrow Y$ e uma ´ relação binária sobre $X \times Y$ que satisfaz:
    - 1- Para cada $x \in X$, existe um $y \in Y$ tal que $[x, y] \in f$.
    - 2- Se $[x, y1] \in f$ e $[x, y2] \in f$, entao˜ $y1 = y2$.
- Função parcial˜ : satisfaz somente a condição 2 acima.

Produto Cartesiano
Operação que constrói um conjunto ´de pares ordenados: $X \times Y = \{[x, y] \mid x \in X \text{ e } y \in Y\}$.

Cardinalidade de conjuntos

- A cardinalidade de conjuntos finitos é a quantidade de elementos que possui;
- Se existe uma bijeção (função bijetora) entre dois conjuntos, ambos possuem a mesma cardinalidade;
- A cardinalidade do conjunto X é´ menor ou igual que a de Y se existe uma função injetiva
- Conjuntos Infinitos
    - Um conjunto é dito infinito se possui um subconjunto próprio com a mesma cardinalidade.
    - Ex: Conjunto dos números naturais e conjunto dos números naturais pares
- Conjunto Universal
    - É o conjunto que possui todos os números n (sendo que o n varia de 0 a infinito -> {n | n > y})

Algumas Notações

- Conjuntos pequenos podem ser definidos explicitamente. Ex: Y = {a, b, c, d, e}.
- Conjuntos infinitos e finitos (com um número grande de elementos) devem ser definidos implicitamente. Ex: {n | n > 9}.
- União : $X \cup Y = \{z \mid z \in X \text{ ou } z \in Y\}$. (união de ambos)
- Interseção: $X \cap Y = \{z \mid z \in X \text{ e } z \in Y\}$. (o'que se repete em ambos)
- Diferença: $X - Y = \{z \mid z \in X \text{ e } z \notin Y\}$. (o'que tem em um e não tem no outro)
- Complemento : $\overline{X} = U - X.$ (retira o conjunto do universal - nI/n>y [y sendo o maior número del conjunto X]).
- DeMorgan: $\overline{(X \cup Y)} = \overline{X} \cap \overline{Y}$ $\qquad$ $\overline{(X \cap Y)} = \overline{X} \cup \overline{Y}.$

Teoremas

- A união de dois conjuntos contáveis é contável.
- O produto Cartesiano de dois conjuntos contáveis é contável.
- O conjunto de subconjuntos finitos de um conjunto contável é contável.

Conjuntos Contáveis e Incontáveis

Conjuntos infinitos contáveis enumeráveis (elementos que podem ser ordenados e contados) possuem a mesma cardinalidade;

- Conjunto contável : conjunto finito ou enumerável ´ .
- Conjunto incontável : conjunto que não é contável (impossível de listar os seus elementos de forma sequencial).

Alguns exemplos:

Argumento de diagonalização de Cantor
Assuma que o conjunto de funções totais de N para N é enumerável, então existe uma sequência f0, f1, f2, . . . que contém todas as funções.
Os valores das funções formam um grid bidimensional.

	0	1	2	3	4	...
f_0	$f_0(0)$	$f_0(1)$	$f_0(2)$	$f_0(3)$	$f_0(4)$	...
f_1	$f_1(0)$	$f_1(1)$	$f_1(2)$	$f_1(3)$	$f_1(4)$	...
f_2	$f_2(0)$	$f_2(1)$	$f_2(2)$	$f_2(3)$	$f_2(4)$	...
f_3	$f_3(0)$	$f_3(1)$	$f_3(2)$	$f_3(3)$	$f_3(4)$	...
f_4	$f_4(0)$	$f_4(1)$	$f_4(2)$	$f_4(3)$	$f_4(4)$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

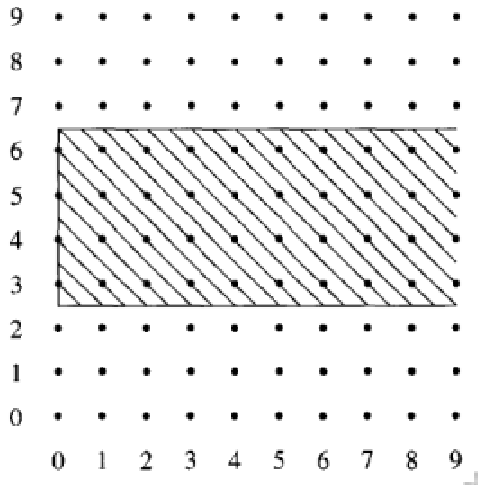
Passos da prova:
- Defina a função f : N → N como f(n) = fn(n) + 1, os valores de f vem da diagonal do grid.
- Pela definição de f, temos que f(i) 6= fi(i) para todo i.
- Portanto, não está na sequência f0, f1, f2, ...
- Contradição! Assumiu-se que a sequência tinha todas as funções totais.
- Assumir que o conjunto de funções totais de N para N é enumerável leva a uma contradição.
- ⇒ O conjunto é incontável. ´

Paradoxo de Russell
Não existe um conjunto de todos os conjuntos existentes, logo, dado um conjunto X ele não pode ser conjunto dele mesmo.
Investiga a relação de pertinência entre dois conjuntos, se dado conjunto é elemento de outro conjunto.
Questiona se o conjunto dos conjuntos que não contêm a si mesmo como membros contêm a si mesmo.

Definições Recursivas
É uma definição recursiva de um conjunto X especifica um método para construir (gerar) os elementos de X.
Possui:

- Base : Princípio pelo qual se parte o problema (objetivo é chegar nesse passo) .
    - Ex: 0 ∈ N.
- Passo Recursivo : Função/ Operações.
    - Ex: se n ∈ N, então s(n) ∈ N.
- Fecho : Formado por todos os elementos gerados a partir da base por um número finito de operações.
    - Ex: n ∈ N somente se n pode ser obtido de 0 por um número finito de aplicações do passo Recussivo.

Exemplos:



Exemplo 1.6.3 (Sudkamp)

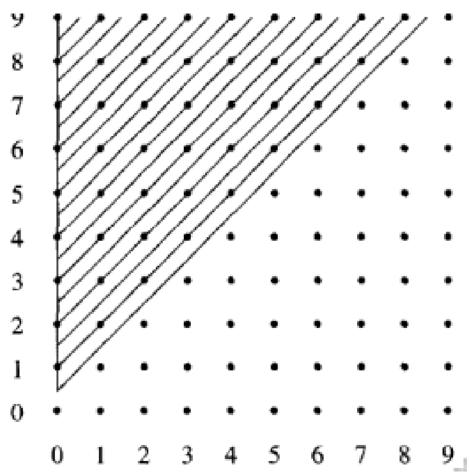
Definição recursiva do conjunto X de valores ao lado.

1. **Base:** $[0,3], [0,4], [0,5], [0,6] \in X$.
2. **PR:** Se $[m, n] \in X$ então $[s(m), n] \in X$.
3. **Fecho**.

Exemplo 1.6.2 (Sudkamp)

Relação LT (*less than*):

1. **Base:** $[0,1] \in LT$.
2. **PR:** Se $[m, n] \in LT$ então $[m, s(n)] \in LT$ e $[s(m), s(n)] \in LT$.
3. **Fecho**.

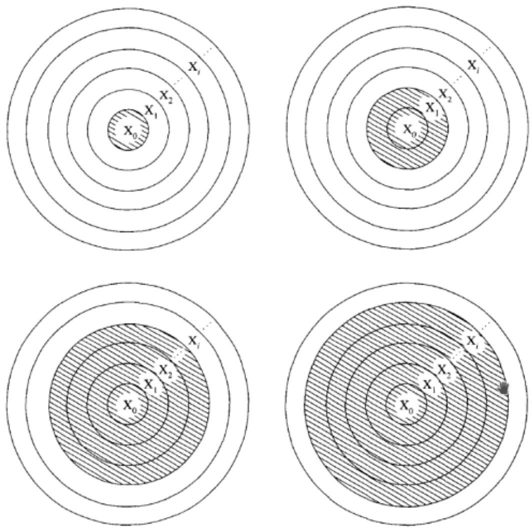


Indução Matemática

Quando se quer provar que uma propriedade P vale para todos os elementos de um conjunto X, se o conjunto for infinito, basta apenas aplicar a propriedade a todos os elementos, entretanto, se for definido recursivamente é necessário utilizar a “Indução Matemática”.

Ex:

- Seja P um propriedade definida sobre os elementos de X. P vale para todos os elementos?
    - Se P vale para o elemento Xi… ele equivale pros elementos Xi + 1, logo P vale para todos os elementos.



- Provar que $n! > 2^n$, para $n \geq 4$.
- **Caso Base** $(n = 4)$: $4! = 24 > 16 = 2^4$.
- **Hipótese Indutiva (HI)**: Assuma que $k! > 2^k$ para $k = 4, 5, \ldots, n$.
- **Passo Indutivo**: Devemos provar que $(n+1)! > 2^{n+1}$.

$$
\begin{aligned}
(n+1)! &= n!(n+1) & \\
&> 2^n(n+1) & \text{(HI)} \\
&> 2^n 2 & \text{(dado que } n+1 > 2) \\
&= 2^{n+1} &
\end{aligned}
$$

Máquina de Turing
É um modelo computacional simples e preciso de computabilidade, que se usa para estudar oque é ou não computável.