

## ***10 CONCLUSÕES***

### **10.1 Resumo dos Principais Conceitos**

### **10.2 Contribuições da Teoria da Computação**

## 10 CONCLUSÕES

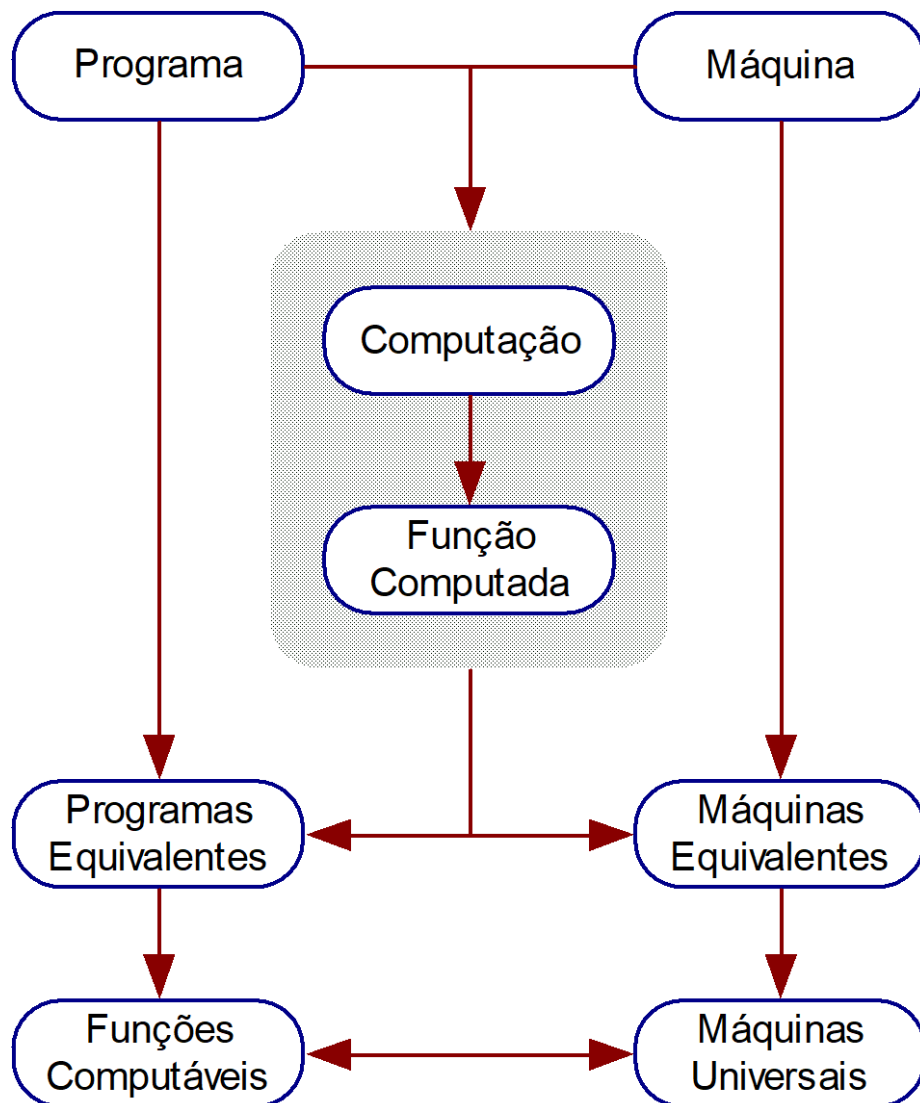
### ➤ A abordagem

- combina abordagens históricas com abordagens próximas dos sistemas computadores modernos.
- objetivo dessa combinação é permitir um fácil entendimento e associação dos problemas abstratos com os problemas típicos da Ciência da Computação atual.

### ➤ O propósito

- **construir, de forma gradual, os diversos conceitos básicos de Teoria da Computação.**
- a noção de procedimento efetivo ou de função computável.  
Para definir as funções computáveis, são descritos diversos formalismos e máquinas.
  - **Máquina de Turing** - é um formalismo simples e usualmente adotado para desenvolver esse tipo de estudo.
  - **Máquina Norma** - possibilita a diferenciação entre programa e máquina, estando, por isso, bastante próxima da noção de computabilidade e dos computadores atuais.
- máquinas universais, nas quais é possível representar qualquer função que seja computável.
- a noção de função recursiva, formalismo equivalente às máquinas universais.
- computabilidade e estudo da solucionabilidade de problemas.
  - **Problemas Solucionáveis** (existe um algoritmo que resolva o problema, para qualquer entrada)
  - **Problemas Não-Solucionáveis** (não existe um algoritmo que sempre resolva o problema).
  - **Problemas Computáveis** (Parcialmente Solucionáveis)
  - **Problemas Não-Computáveis**  
(Completamente Insolúveis).

## 10.1 Resumo dos Principais Conceitos



**Figura 10.1** *Conceitos básicos desenvolvidos*

## ♦ Computabilidade

- Programas
- Máquinas
- Computações
- Função computada

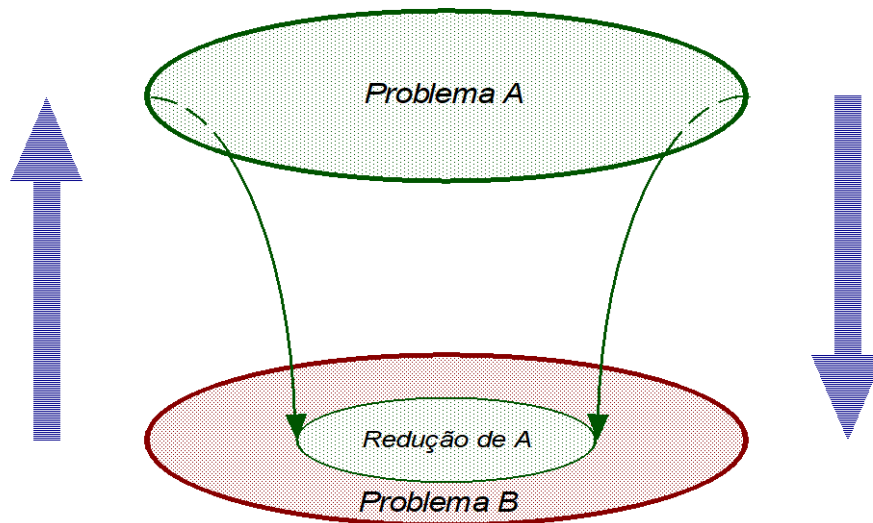
## ♦ Equivalência

- Comparações entre programas
  - Se dois programas, em uma máquina, possuem a mesma função computada, ou seja, computam a mesma função, então eles são equivalentes.
  - Se esses programas são equivalentes em qualquer máquina, então eles são equivalentes fortemente.
- Comparações entre Máquinas
  - Simulação - se uma máquina simula outra, é porque, para qualquer programa da outra máquina, pode-se encontrar um programa desta que faça a mesma coisa.
  - Se duas máquinas simulam-se mutuamente, é porque elas são equivalentes (ambas têm o mesmo poder computacional)
- Qual a máquina mais poderosa?
- Qual o conjunto ou a classe de funções que são computáveis?
  - Uma máquina é universal se toda função computável puder ser executada nela.
  - A Hipótese de Church diz que uma função computável é aquela que pode ser processada numa Máquina de Turing ou equivalente.

## ◆ Solucionabilidade de Problemas

- Princípio da redução

- A idéia básica é investigar a solucionabilidade (respectivamente, não-solucionabilidade) de um problema a partir de outro, cuja solucionabilidade (respectivamente, não-solucionabilidade) é conhecida.



**Figura 10.2 Princípio da Redução**

## 10.2 Contribuições da Teoria da Computação

- proporciona um adequado embasamento teórico necessário para um correto e amplo entendimento da ciência envolvida na computação;
- propicia o desenvolvimento de um raciocínio lógico e formal;
- introduz os conceitos fundamentais que são desenvolvidos em outras áreas

### Linguagens de Programação e Compiladores

- a abordagem de reconhecimento de linguagens é a base de todo o estudo de Linguagens Formais, Semântica Formal, Compiladores e de Linguagens de Programação;

### Orientação a Objetos

- a formalização de objetos através de suas características e propriedades possibilita que se identifiquem classes, as quais podem agrupar esses objetos. Os objetos herdam as propriedades da classe, sendo, portanto, instâncias dessa classe.

### Otimização de Algoritmos;

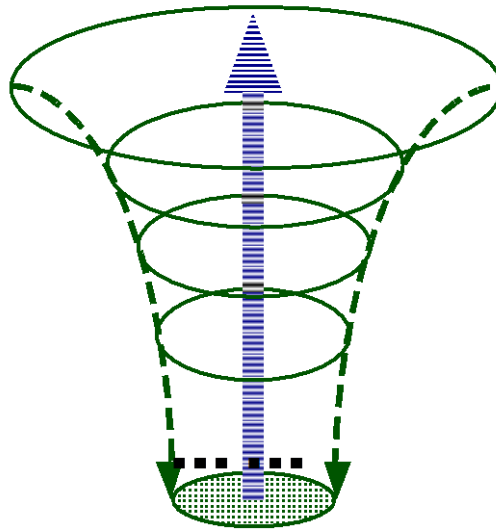
- a abordagem do processamento de funções lança a base para o desenvolvimento de algoritmos eficientes, não só na resolução dos problemas, mas também no estudo da otimização de algoritmos;

### Análise e Desenvolvimento de Algoritmos e Complexidade de Algoritmos Sequenciais e Paralelos;

- A complexidade estrutural diz respeito à estrutura de controle adotada e à otimização em termos do número de instruções e da eliminação de instruções desnecessárias.
- A complexidade computacional diz respeito à quantidade de trabalho envolvida na resolução do problema pelo algoritmo (tempo), medida, muitas vezes, pela quantidade de trabalho que uma determinada instância do problema necessita para resolvê-lo.
- Também se pode considerar a quantidade de memória necessária (espaço) e, no caso de processamento paralelo e distribuído, o número de processadores necessários.

### **Campo Cognitivo**

- proporciona mais um estágio na formação do raciocínio lógico, com destaque ao pensamento indutivo ou recursivo.
  - **aspecto da indução**, parte-se de uma instância base, verifica-se a passagem para instâncias superiores, até construir-se o problema como um todo.
  - **aspecto da recursão**, vê-se o problema como um todo e reduz-se o problema a subproblemas menores, até que se chega a um problema base, iniciando assim o processo reverso, com a resolução de cada instância do problema.



**Figura 10.3 Princípio da Recursão**

- no desenvolvimento de demonstrações e de suas técnicas: o objetivo é capacitar o estudante a desenvolver provas formais e informais.
- no desenvolvimento da capacidade de abstração: propriedades abstratas podem ser especificadas e estudadas independentemente de estruturas.

## 11 LEITURAS SUGERIDAS

- AHO, A.; ULLMAN, J. *Foundations of Computer Science*. New York: W. H. Freeman, 1994. (*Principles of Computer Science Series*).
- ARBIB, M.; KFOURI, A.; MOLL, R. *A Basis for Theoretical Computer Science* Berlin: Springer Verlag, 1981. (*Monographs in Computer Science / The AKM Series in Theoretical Computer Science*).
- BIRD, R. *Programs and Machines: An Introduction to the Theory of Computation (Wiley Series in Computing)*. London: John Wiley & Sons, 1976.
- BRAINERD, W. S.; LANDWEBER, L. H. *Theory of Computation*. New York: John Wiley & Sons, 1974.
- BROOKSHEAR, J. G. *Teoría de la Computación: Lenguajes Formales, Autómatas y Complejidad*. Wilmington: Addison Wesley Longman, 2000. 338p.
- COHEN, D. I. A. *Introduction to Computer Theory*. New York: John Wiley & Sons, 1997. 634p.
- CHURCH, A. *An Unsolvable Problem of Elementary Number Theory*. American Journal of Mathematics, Vol. 58, No. 2, 1936.
- CLARK, K.; COWELL, D. *Programs, Machines and Computation; An Introduction to Theory of Computing*. London: McGraw-Hill, 1976.
- CLARKE, A. C. *3001: A Odisséia Final*. Nova Fronteira, 1997.
- DEDEKIND, R. *Was sind und was sollen die Zahlen?* Braunschweig, 1888.
- EVEY, R. J. *The Theory and Applications of Pushdown Store Machines: mathematical linguistics in machine translation*. Cambridge: Harvard Computation Laboratory, 1963. Report NSF-10.
- FERREIRA, AURÉLIO B. H. *Novo Dicionário da Língua Portuguesa*. Nova Fronteira, 1984.
- GÖDEL, K. *On Formally Undecidable Propositions of Principia Mathematica and Related Systems*. Dover Publications, 1992.
- GÖDEL, K. *On formally undecidable propositions in Principia Mathematica and related systems*. In: The undecidable, M. Davis. Raven Press, 1965, p.4-38.
- HAREL, D. *Algorithmics: The Spirit of Computing*. Addison Wesley, 2004. 3 ed.
- HAREL, D. *Computers Ltd.: What They Really Can't Do*. Oxford University Press, 2003.
- HILBERT, D. *Mathematical Problems - Lecture delivered before the International Congress of Mathematicians at Paris in 1900*. In: Mathematical Developments Arising from Hilbert Problems: Proceedings



- 
- American Mathematical Society, 1976. p.1-34. (Proceedings of Symposia in Pure Mathematics, V. 28 parts 1 & 2).
- HOPCROFT, J.; ULLMAN, J. *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2006. 3 ed.
- JENSEN, K.; WIRTH, N. *Pascal User Manual and Report*. Berlin, Springer-Verlag, 1974.
- KLEENE, S. C. *Representation of Events in Nerve Nets and Finite Automata*. In: Shannon, C. E., McCarthy, J. (Eds.), Automata Studies, Princeton University Press, 1956.
- KNUTH, D. *Art of Computer Programming, Vol. 1*. Reading: Addison-Wesley, 2011. 3 ed.
- LEWIS, H.; PAPADIMITRIOU, C. H. *Elements of the Theory of Computation*. Prentice-Hall, 1997. 2 ed.
- LUCCHESI, C. et al. *Aspectos Teóricos da Computação*. Rio de Janeiro, Instituto de Matemática Pura e Aplicada UFRJ, 1979. 292p. (projeto Euclides).
- MANNA, Z. *Mathematical Theory of Computation*. Dover Publications, 2003.
- MENEZES, P. B. *Linguagens Formais e Autômatos*. Porto Alegre, Bookman, 2010. 6 ed. (Série Livros Didáticos do Instituto de Informática da UFRGS, N.3.)
- MENEZES, P. B. *Matemática Discreta para Computação e Informática*. Porto Alegre, Bookman, 2010. 3 ed. (Série Livros Didáticos do Instituto de Informática da UFRGS, N.19.)
- MENEZES, P. B.; HAEUSLER, E. H. *Teoria da Castegorias para Ciência da Computação*. Porto Alegre, Bookman, 2008. 2 ed. (Série Livros Didáticos do Instituto de Informática da UFRGS, N.12.)
- MENEZES, P. B.; SERNADAS, A.; COSTA, J. *Nonsequential Automata Semantics for a Conccurent Object-Based Language*. In: US-Brazil Joint Workshops on the Formal Foundations of Software Systems, Electronic Notes in Theoretical Computer Science, Vol.14, 1998.URL: <http://www.sciencedirect.com/science/journal/15710661>
- MENEZES, P. B.; TOSCANI, L. V.; LÓPEZ, J. G. *Aprendendo Matemática Discreta com Exercícios*. Porto Alegre, Bookman, 2009. (Série Livros Didáticos do Instituto de Informática da UFRGS, N.16.)
- MINSKY, M. L. *Computation: Finite and Infinite Machines*. Englewood Cliffs: Prentice Hall, 1967.
- OETTINGER, A. G. *Automatic Syntactic Analysis and the Pushdown Store*. In: SYMPOSIA IN APPLIED MATHEMATICS. Vol. 12. Proceedings ... Providence: American Mathematical Society, 1961.
- POST, E. *Finite Combinatory Process: Formulation I*. The Journal of Symbolic Logic, Vol. 1, No. 3, 1936. p.103-105.

- PLUM, T. *Learning to program in C*. Plum Hall, segunda edição, 1989.
- SAGASTUME, M; BAUM, G. *Problemas, Lenguajes y Algoritmos*. Campinas, EBAI/Editora da UNICAMP, 1986. 165p.
- SCHÜTZEMBERGER, M. P. *On Context-Free Languages and Pushdown Automata*. Information and Control, v.6, p.246-264. 1967.
- SERNADAS, C. *Introdução à Teoria da Computação*. Lisboa, Editorial Presença, 1993.
- SIPSER, M. *Introduction to the Theory of Computation*. Course Technology, Boston: PWS Publishing, 2005. 2 ed.
- SHOENFIELD, J. R. *Recursion Theory*. Lecture Notes in Logic, v. 1, Berlin: Springer-Verlag, 1993. 84p. A K Peters/CRC Press, 2001. reimpresso
- STOY, J. F. *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory. Lecture Notes in Logic*. MIT Press, 1981.
- TAYLOR, R. G. *Models of Computation and Formal Languages*. New York: Oxford University Press, 1997.
- TOSCANI, L. V.; VELOSO, P. A. S. *Complexidade de Algoritmos*. Porto Alegre, Bookman, 2008. (Série Livros Didáticos do Instituto de Informática da UFRGS, N.13.) 2 ed.
- TURING, A. M. *On computable numbers, with an application to the Entscheidungsproblem*. Proceedings... London Mathematical Society, 1936.
- WEGNER, P. *Programming Language, Information Structure and Machine Organization*. New York, McGraw-Hill, 1968, 401p.