

UMA PROPOSTA DE ENSINO DOS CONCEITOS BÁSICOS DE COMPUTAÇÃO AOS INGRESSANTES DE UM CURSO DE ENGENHARIA DE COMPUTAÇÃO

Ricardo Luis de Azevedo da Rocha – luis.rocha@poli.usp.br

Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia de
Computação e Sistemas Digitais – PCS
Av. Prof. Luciano Gualberto, trav.3, nº.158
05508-900 – São Paulo – SP

Sidney de Castro – sidcast@uol.com.br

Faculdade de Engenharia da Fundação Santo André, Curso de Engenharia da Computação
Av. Príncipe de Gales, 821
09060-650 – Santo André – SP

Dirceu da Silva – dirceu@unicamp.br

Faculdade de Educação da Unicamp, Departamento de Metodologia do Ensino – Educação,
Ciência e Tecnologia
Rua Bertrand Russel nº 801
13083-865 – Campinas – SP

Resumo: *Ao longo da prática do ensino de algoritmos temos observado um baixo rendimento dos alunos, aliado à dificuldade de extrapolação dos conceitos estudados. Atribuímos este baixo rendimento à necessidade de abstração e à ambigüidade que são inerentes às notações que representam um algoritmo (por exemplo, fluxograma, diagrama de bloco, português estruturado e as linguagens de programação em geral). A dificuldade de abstração é uma característica já estudada, e para enfrentá-la propusemos utilizar a máquina de Turing como notação não apenas para o ensino de algoritmos, mas também como modelo de máquina a ser ensinada ao aluno. Isto porque a máquina de Turing é um modelo de computação suficientemente simples para que possamos usar sua configuração de forma exata. Desta maneira, há uma máquina “concreta” com uma linguagem de descrição dos algoritmos para esta máquina. A ambigüidade e a capacidade de abstração não podem ser eliminadas desta disciplina, mas ficam circunscritas, em termos de simplicidade, à máquina de Turing, que agrega o hardware e o software em uma única notação, de natureza matemática. Esta é uma notação mais familiar ao nosso aluno de graduação. No trabalho são apresentados os pressupostos que fundamentam o ensino e a proposta como um todo.*

Palavras-chave: *Metodologia de Ensino de Conceitos Básicos em Computação, Algoritmos, Máquina de Turing*

1. INTRODUÇÃO

O problema de rendimento dos alunos em qualquer curso de graduação é questão relevante, porém quando ocorre logo no início de um curso de graduação, e nas disciplinas

consideradas formativas, adquire uma dimensão muito mais ampla. Em um curso de Engenharia de Computação as disciplinas introdutórias na linha de Algoritmos e Linguagens de Programação formam a espinha dorsal. A grande maioria das disciplinas técnicas necessita de uma formação adequada nessa linha.

Entretanto, na grande maioria das instituições de ensino em que tivemos a oportunidade de trabalhar ou conhecer, há enorme dificuldade em lidar com o problema. Diversas sugestões de alteração na forma usual de se ensinar esta disciplina foram propostas, embora com resultados modestos. Um exemplo é a proposta de GUZDIAL e SOLOWAY (2002) de se utilizar somente elementos multimídia na apresentação dos problemas algorítmicos. Sugestões desta natureza procuram contemplar em sua forma a atual geração de alunos de graduação, parafraseando os autores citados, a geração “Nintendo”.

A tendência que temos observado é a de reformular modelos para adaptá-los à forma considerada mais moderna, atual, e mais palatável ao aluno da geração “Nintendo”. Estas reformulações, invariavelmente, terminam por propor alterações curriculares simplificadoras, buscando eliminar questões conceituais fundamentais para a formação de um futuro profissional de Computação, porque tais questões apresentam difícil tratamento no formato “videogame”.

O resultado final das mudanças de formato vigentes é, de certa forma, desmotivador, porque desencadeou em conjuntos desconexos de disciplinas, cujo único objetivo é facilitar a compreensão de conteúdos simples e particularizados. Com isso, o aluno perde a noção do todo, a capacidade de generalização e de abstração, já que recebe um conjunto não alinhavado de conteúdos que devem ser transformados por ele em conhecimento. Os princípios, os conceitos fundamentais não são contemplados adequadamente, como bem percebe DENNING (2003), já que eles estão debaixo de camadas de tecnologia, aprisionados. Por outro lado, também distantes do aluno por serem considerados maçantes ou de difícil formatação para a geração “Nintendo”.

A proposta desta pesquisa é trazer à tona, recuperar o papel dos conceitos, dos princípios computacionais. Cada princípio utilizado na computação tem alguma preocupação central, e aspectos técnicos e tecnológicos envolvidos. Nossa proposta busca tornar o fluxo de apresentação dessas questões centrado nos conceitos, sem esquecer de que lidamos com a geração “Nintendo”. Assim, no lugar de apresentar um aspecto e comentar brevemente a origem, o conceito envolvido, propomos voltar a centrar a questão no conceito, e dele derivar para os aspectos.

2. CONTEXTUALIZAÇÃO E DEFINIÇÃO DOS CONCEITOS BÁSICOS

Seguindo o trabalho de DENNING (2003), computação é o conceito mais básico, e a preocupação central é responder à questão do que é computação e do que pode ser computado, ou seja, quais são os limites da computação. Dentro desta perspectiva, os aspectos envolvidos são:

- Algoritmos e estruturas de controle
- Máquinas de Turing
- Linguagens e estruturas de dados
- Computadores universais
- Traduções
- Aproximações
- Não-computabilidade
- Heurística
- Complexidade
- Realizações físicas
- Lógica

Em nosso contexto, há três disciplinas básicas para o aluno ingressante: ‘Introdução à Informática’, ‘Introdução à Engenharia da Computação’, e ‘Fundamentos de Lógica para Computação’. Assim sendo, as questões abordadas na primeira disciplina variam desde Algoritmos até Traduções, a segunda disciplina lida desde Aproximações e Modelos, até Não-

computabilidade, e a terceira disciplina lida com as questões de Lógica e Realizações físicas. As questões relativas à Heurística e Complexidade são brevemente mencionadas nas três disciplinas, e retornam como aspectos fundamentais nas disciplinas da seqüência.

As próximas seções deste artigo descrevem em detalhes a proposta implementada para a disciplina ‘Introdução à Informática’, que cobre a maior parte do conteúdo básico de Computação.

3. USO DE UM MODELO PADRÃO

Para viabilizar a proposta ao público da geração “Nintendo”, nossa primeira preocupação foi encontrar um modelo formal cuja descrição fosse suficientemente simples e ao mesmo tempo abrangente para servir de padrão para descrever e estudar toda a disciplina. Textos já clássicos, como o de HILLIS (2000), descrevem de maneira bastante simples e didática alguns dos modelos básicos para computação. Partindo de modelos lógicos e instigando a curiosidade e a imaginação do leitor, HILLIS (2000) leva-o a descobrir o funcionamento do computador.

Entretanto, o nosso caso é ligeiramente diferente. Precisamos mostrar o que é computação em algum dispositivo, seja ele uma calculadora, um computador, ou ainda um ser humano. Assim sendo, a generalização do conceito de computação precisa estar presente desde o início da disciplina e a particularização deverá ser feita aos poucos. Logo, o modelo buscado deveria ser capaz de compreender todos os demais em termos de capacidade expressiva, e de representação.

Essas características nos levaram a buscar o modelo formal que é a origem do termo e da ciência e engenharia da computação, a máquina de Turing. A proposta tem como padrão, como modelo base, a máquina de Turing e, assim, a partir dela, algoritmos, máquinas, linguagens, etc serão explicados e explicitados.

Partindo da máquina de Turing, expõe-se o conceito de máquina, de dispositivo para computação. Ao mesmo tempo, percebe-se que para cada tipo de problema uma máquina diferente deve ser gerada, embora os elementos básicos estejam presentes. Desta percepção deriva-se diretamente o computador, uma máquina de Turing universal, capaz de simular o funcionamento de qualquer outra máquina de Turing, composto de elementos que permitem armazenar dados e programas, bem como gerar resultados. Deriva-se, ainda, o conceito de algoritmo e de programa de computador através de cada máquina de Turing particular, que é a representação de um programa (algoritmo), descrito em uma linguagem específica.

Observa-se, com a descrição acima, a estreita vinculação entre os conceitos básicos e seu encadeamento, o que ilustra a adequação do modelo escolhido como padrão. A nossa tarefa passou a ser encontrar uma formulação para o modelo que tornasse a sua descrição mais próxima da geração “Nintendo” e, por outro lado, que facilitasse a compreensão de conceitos tão abstratos. O artigo de CHESÑEVAR, COBO e YURCIK (2003), que compara simuladores para aplicação didática de modelos computacionais, apontou o caminho: há um simulador didático e propício para a geração de alunos com os quais trabalhamos - o VTuring¹. O uso deste simulador permite tornar menos abstrata a apresentação de conceitos como computação e, ao mesmo tempo, é muito propício para o uso de alunos da geração “Nintendo”. O simulador passa a ser visto como uma máquina “concreta”, “palpável”, quase “física”.

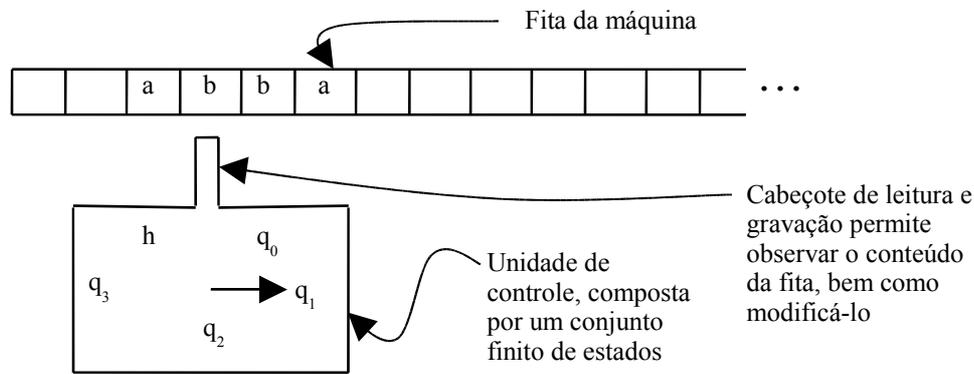
3.1 Máquina de Turing

O matemático inglês Alan Turing propôs, em 1936, um dispositivo capaz de responder à questão da computação. Este é um dispositivo teórico (base da teoria da computação), e funciona utilizando idéia de apresentar uma resposta a um problema através da utilização de uma seqüência de funções, dividindo o processo em pequenas ações. O texto a seguir é uma descrição simplificada de LEWIS e PAPADIMITRIOU (2000).

A máquina de Turing possui um dispositivo de entrada, que é representado por uma fita de capacidade infinita à direita e limitada à esquerda, dividida em setores (células), enumeráveis e endereçáveis. Possui ainda um cabeçote de leitura e gravação, que permite à máquina ler e escrever na fita. O controle das funções da máquina é obtido através de uma

¹ Pode ser encontrado em www.cheransoft.com/vturing

unidade especial, que trabalha como uma máquina de estados finita. Esquemáticamente tem-se a seguinte figura:



A fita da máquina é dividida em células de mesmo tamanho, que armazenam somente um símbolo (um elemento). Os valores encontrados na fita são definidos pelo alfabeto da máquina de Turing. Este alfabeto inclui um símbolo especial (“#”) para marcar uma célula sem símbolos (espaço em branco).

O alfabeto de uma máquina de Turing é composto por um conjunto de símbolos, os quais podem ser compostos (agrupados) de modo a formar cadeias (“strings”). Uma cadeia que se encontra na fita de uma máquina de Turing, antes do início de sua execução, é chamada de cadeia de entrada, e a cadeia produzida pela máquina ao fim de sua execução, cadeia de saída.

O cabeçote de leitura e gravação pode mover-se para a esquerda ou para a direita, de acordo com o estado da máquina, apenas uma célula por vez. Isto significa que, para a máquina atingir uma célula, localizada a duas células da célula corrente (no qual o cabeçote está parado), a máquina deve partir da célula corrente e andar duas células, uma em cada movimento do cabeçote (o que pode significar mudar de estado na unidade de controle).

A unidade de controle é composta de um conjunto de estados, sendo importante observar que há dois estados especiais - o estado inicial e o estado final - e uma função de transferência. Um estado representa uma situação da máquina de Turing, no qual, um conjunto de ações pode ser tomado de acordo com a função de transferência e também do símbolo encontrado na fita.

A função de transferência da máquina de Turing mapeia as ações e os estados alcançáveis a partir do estado corrente e do símbolo encontrado na fita (é responsável por trocar o estado da máquina, a partir do estado atual e do símbolo encontrado na fita). Funciona de forma similar a uma tabela de transições entre estados, na qual para cada estado tem-se o próximo estado possível, a partir do símbolo encontrado. Na troca de estado, caso a função de transferência mapeie a partir do estado atual e do símbolo na fita, o estado especial final (h - “halt”, parada), a máquina termina sua execução e fica parada.

Assim com os símbolos do alfabeto da máquina de Turing e as possíveis ações da máquina (movimenta uma posição à direita, movimenta uma posição à esquerda, e escreve símbolo na fita na posição do cabeçote), podemos reduzir o tradicional ensino de algoritmos a uma aplicação direta a uma arquitetura de dispositivo computacional (computador).

4. O PROCESSO

A atividade do aluno passou a ser manipular cadeias de caracteres que representem as soluções. Esta simplificação, entretanto, não eliminou a apresentação dos conceitos básicos de computação. Para ilustrar, temos o exemplo de um aluno que produziu o seguinte comentário: “Professor, este game (o Vturing) eu ainda não consegui dar fim, e acho que não vai ter fim nunca, mas posso fazer qualquer coisa que minha imaginação quiser só com estas regrinhas tão básicas!”. Embutida na frase típica da geração “Nintendo” está a compreensão do que é computação e, mais do que isso, demonstra que o aluno é capaz de criar qualquer ação usando um misto de funções de hardware e software.

Para atingir este estágio, devemos propiciar um ambiente mais adequado do que o uso direto do VTuring, definindo uma função genérica de entrada de dados, e outra de saída de dados. Desta maneira, o aluno percebe o ambiente como um computador mais próximo da

realidade, e a linguagem de programação como uma linguagem de montagem. Em nosso caso, definimos:

- **stin**: função de entrada da máquina, com dois parâmetros o registrador e o valor do dado;
- **registradores**: três ao todo (representados em nosso caso por “u”, “d”, e “s”) cada registrador tem 8 células ao todo.

Com essas definições, os problemas passam a ser puramente algorítmicos, tanto no sentido formal, quanto na percepção do aluno. Para tornar mais clara esta visão, o exemplo a seguir a ilustra.

4.1 Exemplo Ilustrativo

Sendo fornecidos dois números em alfabeto binário da seguinte forma: **stin(u, "01010000")** e **stin(d, "00011011")**, crie uma função que os compare e apresente como resultado um dos símbolos: “u”, se o valor no registrador **u** for maior do que o valor no registrador **d**; “d”, caso o valor em **d** seja maior, ou ainda, “i”, caso os valores sejam iguais.

Para o aluno, este é um problema cuja solução passa a ser visualizar uma máquina de Turing com a seguinte configuração na fita: #01010000u00011011d#. E este problema admite algumas soluções, como, por exemplo, a descrita a seguir:

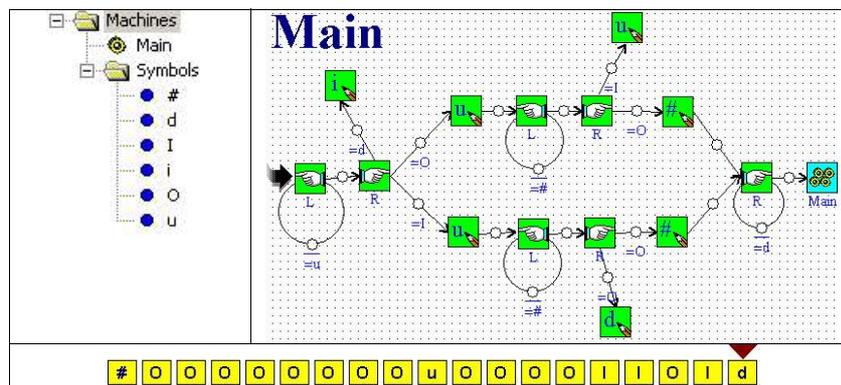


Figura 1 – Uma possível solução para o problema do exemplo.

O aluno consegue visualizar esta máquina com alguma facilidade e é capaz de expor a solução em forma textual, na “linguagem de máquina”, da seguinte forma:

Tabela 1 – Solução encontrada a partir da figura 1.

Q / ΣUF	0	1	u	d	i	#	Descrição da ação
q0	q0,L	q0,L	q1,R	q0,L	q0,L	q0,L	Caminha para a esquerda enquanto diferente de “u”.
q1	q2,u	q5,u	**	**	halt,i	**	Ao encontrar “d” ambos serão iguais, separa o caminho para o caso zero e o caso um marcando a casa visitada na primeira seqüência.
q2	q2,L	q2,L	q2,L	**	**	q3,R	Caminha para a esquerda enquanto diferente de “d”.
q3	q4,#	halt,u	**	**	**	**	Partindo de zero na primeira seqüência e encontrado um na segunda, a segunda será a maior então “u”.
q4	q4,L	q4,L	q4,L	q1,R	q4,L	q4,L	Retorna para o início do processo.
q5	q5,L	q5,L	q5,L	**	**	q6,R	Caminha para a esquerda enquanto diferente de “d”.
q6	q6,#	halt,d	**	**	**	**	Partindo de 1 na primeira seqüência e encontrado 0 na segunda, a segunda será a maior então “d”.

5. CONSIDERAÇÕES FINAIS

Esta proposta foi implantada em uma instituição pública de direito privado de ensino superior na região do ABC, no Estado de São Paulo. A condução desta experiência foi bastante proveitosa e os resultados, embora ainda não finalizados, apontam para uma maior adequação e participação dos alunos. Isto se deve, basicamente, pelo fato de o aluno não se

sentir um elemento passivo. Ao contrário, ele é o agente de sua capacitação, afinal ele dita o próprio ritmo.

No nosso caso, esta é uma característica extremamente desejável, porque lidamos com classes cuja quantidade de alunos dificulta o atendimento individualizado em um processo tradicional. Embora seja importante haver acompanhamento próximo, através de monitoria, laboratórios, etc.

Na tentativa de responder de maneira mais eficiente (e muitas vezes mais rápida) às questões formuladas, os alunos não se basearam em soluções ortodoxas. Procuraram criar, enfim, tiveram ousadia na apresentação de suas respostas. Isto é possível devido ao modelo adotado, extremamente simples, com poucas instruções (na verdade três), e um alfabeto cuja quantidade de símbolos é definida de acordo com o problema.

O fato de termos usado extensivamente a representação de símbolos em um alfabeto binário possibilitou a abstração de problemas da realidade, como por exemplo, números em uma nota fiscal, valor da nota, etc. E tal característica foi percebida como positiva pelos alunos, ilustrando que, ao menos, foi possível avançar um pouco na questão da dificuldade de abstração.

A chance de testar a sua possível solução sem interferência de outros elementos condicionantes, relativos à linguagem, ao laboratório, ao monitor, etc, garante que o aluno tenha uma oportunidade de expressar seus anseios mais facilmente, fazendo com que ele ganhe autonomia e possa exibir alto grau de ousadia.

O aluno passa a ter uma visão mais completa da solução e tem a possibilidade de visualizar aplicações práticas dos problemas em situações reais, já que a sua solução, bem como o problema original, estão em um nível de abstração maior. Outro aspecto é a percepção dos limites de computação, não apenas pelo fato da máquina estar limitada, mas de haver problemas que sequer podem ser modelados como um processo computacional. Assim sendo, o famoso problema da parada da máquina de Turing torna-se menos abstrato, embora seja mencionado sem demonstração nesta disciplina.

Ainda com relação aos limites, o próprio aluno passa a ter condições de perceber como determinadas soluções são piores do que outras, seja por se tornarem mais lentas, ou por ocuparem mais recursos da máquina.

Finalizando, ao final desta disciplina o aluno passa a ter uma visão adequada da computação como um processo, sem mistérios ou visões deturpadas, quase místicas. Compreende, em parte, a importância dos aspectos formais e das disciplinas de base matemática.

REFERÊNCIAS BIBLIOGRÁFICAS

CHESÑEVAR, C.I.; COBO, M.L.; YURCIK, W. Using Theoretical Computer Simulators to Formal Languages and Automata Theory. In: SIGCSE'2003, 2003, Reno, Nevada. **Proceedings**. ACM, 2003. disponível em: <http://cs.uns.edu.ar/~cic/publications.htm>.

DENNING, P.J. Great Principles of Computing. **Communications of the ACM**. v. 46, n. 11, p. 15-20, 2003.

GUZDIAL, M.; SOLOWAY, E. Teaching the Nintendo Generation to Program. **Communications of the ACM**. v. 45, n. 4, p. 17-21, 2002.

HILLIS, D. **O padrão gravado na pedra**. Editora ROCCO, 2000.

LEWIS, H.R.; PAPADIMITRIOU, C.H. **Elementos de Teoria da Computação**. Porto Alegre: Bookman, 2000.

A PROPOSAL FOR TEACHING BASIC CONCEPTS OF COMPUTATION TO FRESHMEN UNDERGRADUATE STUDENTS OF COMPUTATION ENGINEERING

Abstract: Throughout the practice of teaching algorithms we have observed that students have presented low performance and difficulty in extrapolating the concepts that have taught. We believe this low performance is due to the necessity of abstraction and the ambiguity related to the notation that represents a certain algorithm (for instance, flowchart, block diagram, pseudo language and programming languages in general). The difficulty of abstraction is a characteristic that has already been studied and, to confront it, we recommend the use of the Turing machine as a notation, not only to teach algorithms, but also as a machine model to be taught to the student. Because the Turing machine is a computational model simple enough so that we can use its configuration in its exact form. This way, there is a “concrete” machine with an algorithm description language to it. The ambiguity and capacity of abstraction cannot be removed, but they are circumscribed, in terms of simplicity, by the Turing machine, which unites hardware and software in one single notation, with mathematical nature. This is a more familiar notation to our undergraduate student. In this paper, the underlying concepts that formed the basis for the teaching and the proposition as a whole are presented.

Key-words: Teaching Methodology of Basic Concepts of Computation, Algorithms, Turing Machine