Lógica Computacional

Flávio L. C. de Moura e Rafael Monteiro Rodrigues
Departamento de Ciência da Computação
Universidade de Brasília[1]

[1]flavio@flaviomoura.info

Introdução e Motivação

O que é Lógica Computacional?

A *Lógica Computacional* (LC) tem por objetivo utilizar a lógica para raciocinar sobre Computação, ou seja, consiste na utilização da lógica para a resolução de problemas computacionais. Este conceito pode ser utilizado de diversas formas, por exemplo, é comum a associação da LC com programação em lógica. Neste caso, a lógica é utilizada como uma linguagem de programação [1]. Uma outra abordagem possível é a simples mecanização do raciocínio lógico, de forma a permitir a resolução de exercícios de lógica no computador, ao invés da resolução usual em papel e lápis [2]. A abordagem que utilizaremos difere das anteriores, mas possui um pouco de cada uma delas como veremos a seguir.

Para explicar a nossa abordagem, suponha que você tenha um grande banco de dados com informações de uma determinada população, e que por alguma razão precise ordenar estas informações por idade em determinado momento, em outro, a ordenação que precisa ser feita é por nome ou outro critério qualquer. O que você faz? Uma alternativa é utilizar alguma implementação já feita e resolver o problema. Uma pergunta que pode ser feita é: será que a implementação utilizada gera a resposta correta? Outra alternativa seria construir/implementar um algoritmo de ordenação, mas a questão sobre a correção ainda continuaria válida: a implementação construída é correta? A abordagem que utilizamos neste curso fornece as ferramentas necessárias para responder a estas perguntas. Em particular, estudaremos sistemas dedutivos que nos permitirão provar propriedades de programas [3].

A Correção de programas

Obviamente desejamos utilizar apenas programas de computador que sejam corretos, *i.e.* programas que fazem aquilo que se espera que ele faça... mas como saber isto? Existem diversos exemplos famosos de erros em sistemas computacionais dentre os quais destacamos:

1. **Pentium FDIV**: Um erro na construção da unidade de ponto flutuante do processador Pentium da Intel causou um prejuízo de aproximadamente 500 milhões de dólares para a empresa que se viu forçada a substituir os processadores que já estavam no mercado em 1994.

2. **Therac-25**: Uma máquina de radioterapia controlada por computador causou a morte de pelo menos 6 pacientes entre 1985 e 1987 por overdose de radiação.

3. **Ariane 5**: Um foguete que custou aproximadamente 7 bilhões de dólares para ser construído explodiu no seu primeiro voo em 1996 devido ao reuso sem verificação apropriada de partes do código do seu predecessor.

Uma abordagem possível, e bastante comum, para tentar responder se um dado programa é correto consiste na realização de testes. De fato, a primeira coisa que fazemos após implementar um algoritmo é testá-lo para algumas entradas possíveis. Caso alguma saída esteja fora do esperado, uma revisão da

implementação é feita para corrigir o erro, e então novos testes são realizados. Este processo é repetido até que o programador sinta confiança na implementação, mas a pergunta persiste: depois de todos estes testes, é possível dizer que o programa é correto? Certamente não... pensando no caso particular da implementação de um algoritmo de ordenação de inteiros, existe uma infinidade de listas de inteiros que podem ser utilizadas nos testes, e portanto não é possível testar todas elas. O que fazer então? Uma alternativa é utilizar a lógica para **provar** que o programa é correto. Uma prova matemática de uma propriedade de um programa fornece a garantia de que o programa satisfaz a propriedade provada **sempre**! Esta é a abordagem que utilizaremos no curso. Mas o que é uma prova? Alguém pode dizer que "é um argumento feito para convencer alguém"[4]. O problema desta definição é que pessoas diferentes podem ter compreensões distintas sobre o argumento, de forma que o argumento seja uma prova para uma, mas não para a outra... estranho, não? Uma definição geral e abstrata para a noção de prova não é uma tarefa fácil, mas forneceremos uma definição precisa em um contexto mais restrito, a saber, o da lógica simbólica.

A Lógica Proposicional

Antes de definirmos a *Lógica Proposicional* (LP) é natural nos perguntarmos: o que é lógica? Considere a definição apresentada em [5]:

> A lógica é a ciência que estuda princípios e métodos de inferência, tendo como objetivo principal determinar em que condições certas coisas seguem (são consequência), ou não, de outras.

Ao longo do curso detalharemos alguns dos conceitos que podem não estar claros nesta definição, como por exemplo, o de *inferência*, *consequência* e *coisas que seguem*. De qualquer forma, utilizamos a lógica, mesmo que de maneira informal, sistematicamente em nosso dia a dia. Por exemplo, João disse que morou 5 anos na França, então podemos concluir que ele fala francês. Esta conclusão (João fala francês) é uma consequência de um fato conhecido (João disse que morou 5 anos na França). Mas por que estudar lógica? E em particular, lógica computacional? Para responder a esta pergunta, vamos considerar o seguinte problema:

- **Exercício** (Adaptado de [4]) Considere uma ilha onde moram apenas dois tipos de pessoas: as honestas, e que portanto sempre falam a verdade; e as desonestas, que sempre mentem. Um viajante, ao passar por esta ilha encontra três moradores chamados A, B e C. O viajante pergunta para o morador A: "Você é honesto ou desonesto?" A responde algo incompreensível, e o viajante pergunta para B: "O que ele disse?" B então responde "Ele disse que é desonesto". Neste momento C se manifesta: "Não acredito nisto! Isto é uma mentira!". Questão: C é honesto ou desonesto?